

# Proyecto Trabajo Fin de Grado

## Grado en Ingeniería en Tecnologías Industriales

### Simulación cinemática y dinámica de un vehículo controlado por control remoto en tres dimensiones

Autor: Alberto González González

Tutor: Juana María Mayo Nuñez

**Dpto. Ingeniería Mecánica y Fabricación**  
**Escuela Técnica Superior de Ingeniería**  
**Universidad de Sevilla**

Sevilla, 2018







Proyecto Trabajo Fin de Grado  
Grado en Ingeniería en Tecnologías Industriales

# **Simulación cinemática y dinámica de un vehículo controlado por control remoto en tres dimensiones**

Autor:

Alberto González González

Tutor:

Juana María Mayo Nuñez

Catedrática

Dpto. de Ingeniería Mecánica y Fabricación

Escuela Técnica Superior de Ingeniería

Universidad de Sevilla

Sevilla, 2018



Proyecto Trabajo de Fin de Grado: Simulación cinemática y dinámica de un vehículo controlado por control remoto en tres dimensiones

Autor: Alberto González González

Tutor: Juana María Mayo Nuñez

El tribunal nombrado para juzgar el Proyecto arriba indicado, compuesto por los siguientes miembros:

Presidente:

Vocales:

Secretario:

Acuerdan otorgarle la calificación de:

Sevilla, 2018

El Secretario del Tribunal



*A mi familia*

*A mis maestros*



# Agradecimientos

---

Me gustaría agradecer a D<sup>a</sup> Juana María Mayo Nuñez por darme la oportunidad de realizar el presente trabajo de fin de grado con ella, así como por haberme ayudado y guiado a lo largo de todo el proceso.

A la par me gustaría agradecer a mi familia, por el apoyo que he recibido durante toda la carrera ya no habría sido posible si no fuera por ellos.





El presente trabajo de fin de grado tiene como finalidad la simulación cinemática y dinámica de un vehículo motorizado por control remoto. El punto de partida de este proyecto nace de las prácticas desarrolladas en la asignatura de Cinemática y Dinámica de Máquinas, durante las cuales se realizó la simulación del vehículo que describimos en este proyecto, con la singularidad de haber sido realizado en dos dimensiones.

En este caso, se ha realizado la simulación en tres dimensiones, introduciendo nuevas coordenadas al sistema que permitían el movimiento del vehículo sobre cualquier terreno, siempre que éste no se despegue del suelo. La introducción de dichas coordenadas ha supuesto en paralelo la incorporación de nuevas ecuaciones de restricción que garantizaran la resolución del sistema propuesto.



# Abstract

---

The current Project aims to simulate the kinematics and dynamics behaviour of a remote control vehicle. The starting point for this Project is the practicum done over the subject “Cinemáticas y Dinámicas de Máquinas”, whose goal was to simulate the behaviour of this vehicle on a two dimensional system.

Nevertheless, its been done the simulation on a three dimensional system, entering new coordinates that allows the vehicle to work at any road as long as its wheels do not lift off the ground. Due to this new coordinates, its been entered new constraints ecuations as well that enables the system resolution.

Agradecimientos

Resumen

Abstract

Índice de tablas

Índice de figuras

Notación

## **1. Introducción**

*1.1. Objetivos*

*1.2. Estructura*

## **2. Sistemas multicuerpo**

*2.1. Introducción*

*2.2. Tipos de coordenadas*

*2.3. Problema de posición*

*2.4. Simulación cinemática*

*2.5. Simulación dinámica*

## **3. Cinemática y dinámica del vehículo**

*3.1. Modelo Completo*

3.1.1. Selección de coordenadas

3.1.2. Matrices de orientación

3.1.3. Posición y velocidades de los sólidos

3.1.4. Ecuaciones de restricción

3.1.4.1. Rodadura sin deslizamiento

3.1.4.2. Contacto rígido

3.1.5. Dinámica

*3.2. Modelo simplificado*

3.2.1. Selección de coordenadas

3.2.2. Matrices de orientación

3.2.3. Posición y velocidades de los sólidos

3.2.4. Ecuaciones de restricción

3.2.4.1. Rodadura sin deslizamiento

3.2.4.2. Contacto rígido

3.2.5. Dinámica

*3.3. Superficie parametrizada*

### *3.4. Parámetros vehículo*

## **4. Programas utilizados para la simulación**

### *4.1. Cálculo simbólico*

### *4.2. Simulación dinámica*

### *4.3. Animación*

### *4.4. Registro de Datos*

### *4.5. Funciones adicionales*

### *4.6. Análisis programa de simulación dinámica*

#### *4.6.1. Modelo simplificado*

#### *4.6.2. Modelo completo*

## **5. Resultados experimentales de la simulación**

### *5.1. Registro de señales*

### *5.2. Tratamiento de señales*

### *5.3. Superficie parametrizada*

### *5.4. Resultados*

## **6. Conclusiones y desarrollos futuros**

### *6.1. Conclusiones*

### *6.2. Desarrollos futuros*

## **7. Anexos**

### *7.1. Anexo A. Cálculo Simbólico.*

### *7.2. Anexo B. Simulación Dinámica Inversa*

### *7.3. Anexo C. AnimaRover*

### *7.4. Anexo D. Registro de Datos*

## **8. Bibliografía**

# ÍNDICE DE TABLAS

---

**Tabla 1.** Comparativa coordenadas y ecuaciones en el modelo completo.

**Tabla 2.** Comparativa coordenadas y ecuaciones del modelo simplificado.

**Tabla 3.** Error ángulo trayectoria vertical respecto ángulo carretera.

**Tabla 4.** Errores ángulo cabeceo respecto ángulo carretera.

# ÍNDICE DE FIGURAS

---

- Figura 1.** Ángulo de Euler.
- Figura 2.** Modelo Rover en 3D.
- Figura 3.** Dimensiones características del Rover.
- Figura 4.** Superficie parametrizada.
- Figura 5.** Trayectorias verticales modelo simplificado.
- Figura 6.** Ángulos de cabeceo modelo simplificado.
- Figura 7.** Trayectorias sobre plano XY modelo simplificado.
- Figura 8.** Trayectorias sobre plano XY modelo completo.
- Figura 9.** Trayectorias verticales modelo completo.
- Figura 10.** Ángulos de cabeceo modelo completo
- Figura 11.** Ángulos de balanceo modelo completo
- Figura 12.** Trayectoria vertical con tres tramos a pendientes diferentes
- Figura 13.** Rover y control remoto
- Figura 14.** Ángulos girados por las ruedas
- Figura 15.** Velocidades de las ruedas
- Figura 16.** Aceleraciones de las ruedas
- Figura 17.** Rampa utilizada para la simulación
- Figura 18.** Trayectoria vertical experimental
- Figura 19.** Ángulo cabeceo experimental
- Figura 20.** Ángulo balanceo experimental
- Figura 21.** Trayectoria sobre el plano XY experimental
- Figura 22.** Velocidades angulares de las ruedas
- Figura 23.** Velocidad angular IMU frente a la calculada
- Figura 24.** Aceleración IMU frente calculada





# Notación

---

XYZ	Sistema de coordenadas globales
$q$	Coordenadas generalizadas
$\dot{q}$	Velocidades generalizadas
$\ddot{q}$	Aceleraciones generalizadas
$A^{-1}$	Inversa de la matriz A
$A^T$	Traspuesta de la matriz A
dep	dependientes
ind	independientes
C	Ecuaciones de restricción
$C_q$	Jacobiano de las ecuaciones de restricción holónomas
$C_t$	Derivada parcial de las ecuaciones de restricción holónomas con respecto al tiempo
B	Jacobiano de las ecuaciones de restricción no holónomas
D	Matriz de los jacobianos de las ecuaciones de restricción no holónomas y holónomas
E	Matriz de las derivadas parciales de las ecuaciones de restricción no holónomas y holónomas con respecto al tiempo
$\dot{f}(x)$	Derivada parcial de la función f respecto a x
$\dot{f}(x)$	Derivada temporal de la función f
$F^i$	Fuerzas aplicadas sobre el sólido i
$m^i$	Masa del sólido i
$r^i$	Posición del centro de gravedad del sólido i
$v^i$	Velocidad del centro de gravedad del sólido i
$a_G^i$	Aceleración del centro de gravedad del sólido i
$\bar{x}$	Vector x en coordenadas locales
$\bar{M}_i$	Momentos aplicados sobre el sólido i
$\bar{L}$	Momento cinético
$\bar{\omega}$	Velocidad angular
$\bar{\alpha}$	Aceleración angular
$\bar{I}_G$	Tensor de inercias aplicado en el centro de gravedad
$R_f$	Radio ruedas delanteras
$R_r$	Radio rueda trasera
$t_{cx}$	Tangente sobre la superficie de la carretera en dirección x
$t_{cy}$	Tangente sobre la superficie de la carretera en dirección y
$ x $	Módulo de x
$\times$	Producto escalar
sen	Función seno

cos

Función coseno

# 1 INTRODUCCIÓN

---

La cinemática y dinámica de vehículos trata con las ecuaciones que permiten describir los movimientos realizados por los vehículos considerando las fuerzas que los producen. Mediante la simulación se pueden obtener estos movimientos en instantes sucesivos de tiempo.

El uso de la simulación de vehículos ha ganado importancia a lo largo de los años debido a la gran competencia existente actualmente dentro de la industria automovilística, así como los complejos procesos de calidad que cada vehículo tiene que superar con el objetivo de poder ser puestos a la venta. Dado estas condiciones, la industria automovilística invierte en la creación de sistemas de software para la simulación de prototipos con los que consiguen reducir en gran parte los costes de prueba y error. [1]

El origen de este trabajo se sitúa en las prácticas de la asignatura de Cinemática y Dinámica de máquinas en las que realizamos la simulación de un vehículo motorizado, conocido con el nombre de Rover, con la singularidad de ser en el caso plano. El procedimiento propuesto para su desarrollo constaba de tres fases: a) procesado de los datos obtenidos por el conjunto de sensores y la identificación del centro de gravedad del vehículo, b) simulación cinemática del vehículo y c) simulación dinámica inversa junto con una serie de comprobaciones que pusieran en relación los datos obtenidos con los sensores con los valores de la simulación dinámica.

En el modelo de dos dimensiones explicado con anterioridad, las únicas ecuaciones de restricción de las que disponía el sistema eran las ecuaciones debido a la condición de no deslizamiento de las ruedas, siendo éstas restricciones no holónomas. Este conocimiento de base nos ha servido para el desarrollo del modelo de tres dimensiones propuesto; incluyendo un mayor número de coordenadas y de ecuaciones de restricción, manteniendo el mismo número de grados de libertad, es decir, los datos de entrada a nuestro problema, siendo en este caso los ángulos girados por las ruedas motoras.

Las limitaciones de nuestro trabajo residen en los grados de libertad de los que dispone el Rover, y de las limitaciones que nos proporcionan el número de sólidos y las uniones rígidas entre sólidos, ya que el uso de un vehículo con un mayor número de sólidos y uniones no rígidas entre ruedas y chasis nos permitiría obtener resultados cuyo comportamiento se asimilará mejor al de un vehículo actual. Por ello, para futuros trabajos se propone la elaboración de un Rover con cuatro ruedas y un eje de giro de las ruedas independientes al chasis.

## 1.1 Objetivos

El objetivo principal que se pretende conseguir con este Trabajo de Fin de Grado es la simulación cinemática y dinámica del comportamiento dinámico del rover en tres dimensiones. Para ello, se va a hacer uso de las ecuaciones utilizadas para la simulación de sistemas multicuerpos y se van a analizar los resultados obtenidos en la simulación con los trayectos realizados experimentalmente.

Tanto los parámetros dimensionales como los parámetros de masa, inercia y centros de gravedad de cada uno de los sólidos han sido obtenidos del estudio previo realizado en la asignatura descrita con anterioridad, de forma que la explicación de cómo se han obtenido esos resultados no forma parte de los objetivos del presente trabajo.

La simulación y los cálculos correspondientes se van a realizar con el programa de cálculo matemático Matlab

versión R2016a.

## **1.2 Estructura**

En este trabajo se presenta en primer lugar un bloque teórico, en el que se abordan los conceptos fundamentales sobre sistemas multicuerpos utilizados para la resolución del problema.

A continuación, se desarrolla la cinemática y dinámica de los distintos modelos utilizados en el Rover, un modelo completo y un modelo simplificado, basándose en las ecuaciones descritas en el bloque anterior.

Con mayor concreción en el tercer capítulo se describen los programas generados para la simulación junto con un análisis de los resultados analíticos obtenidos antes de la experimentación con el Rover que tiene como objetivo servir como comprobación para asegurar los resultados experimentales.

Finalmente un último capítulo en el que se presenta el tratamiento de los datos registrados y los resultados de la experimentación realizada con el Rover. Los resultados obtenidos nos permiten establecer las conclusiones del trabajo desarrollado así como plantear en un sentido prospectivo nuevos desarrollos dentro de este campo.

## 2 SISTEMAS MULTICUERPO

---

### 2.1 Introducción

Un sistema multicuerpo puede definirse como un conjunto de sólidos rígidos o deformables unidos entre sí por pares cinemáticos y actuadores, permitiendo el movimiento relativo entre ellos.[2] Esta definición también podría aplicarse a la de un mecanismo, sin embargo, en este caso es utilizada para referirse a sistemas de mayor complejidad como puede ser un propio vehículo.[3]

Los sólidos rígidos de un sistema multicuerpo se caracterizan por su masa y momento de inercia, y los puntos característicos son su centro de gravedad y los puntos en los que fuerzas o momentos están aplicados, así como los puntos de unión con el resto de los sólidos. [4]

Los algoritmos y ecuaciones utilizados para la resolución están pensados para ser resueltos computacionalmente, en caso de no ser así, éstas ecuaciones cambiarían totalmente. A su vez, a la hora de entender el desarrollo de las ecuaciones es importante conocer que el funcionamiento de un ordenador difiere con el de un cerebro humano y que el éxito de la resolución de este sistema reside en que el proceso sea sistemático.[3]

### 2.2 Tipos de coordenadas

La selección del tipo de coordenadas es un parámetro esencial en términos de la sistematicidad de la que van a disponer nuestros algoritmos. Por ello, podemos definir los siguientes tipos de coordenadas:

- *Coordenadas Lagrangianas*: Son un conjunto de magnitudes geométricas, distancias o ángulos que indican la posición u orientación de cada uno de los sólidos con respecto de la barra fija. [3]
- *Coordenadas relativas*: La posición u orientación de los sólidos están definidos respecto al miembro contiguo, de forma que conocida las coordenadas de uno de ellos respecto al sistema global, podemos obtener la posición del resto de sólidos de nuestro sistema. Por tanto, se puede decir que las coordenadas relativas están asociadas a los pares cinemáticos del sistema.[3]
- *Coordenadas de referencia*: Consiste en la identificación del centro de gravedad de cada uno de los sólidos con respecto al sistema global XYZ y la orientación de cada uno de los ejes locales con respecto al sistema global. De forma que tendríamos 6 coordenadas (3 posiciones y 3 ángulos) por cada sólido en el caso en tres dimensiones, y 3 coordenadas en el caso plano (2 posiciones y 1 ángulo)
- *Coordenadas naturales*: Surgen como evolución de las coordenadas de referencia en las que en lugar de ser introducidas en los centros de gravedad de cada sólido, éstas coordenadas son fijadas con la posición de los puntos característicos del sistema, de forma que queda perfectamente definida la posición y orientación de todos los sólidos. [5]

Además de esta clasificación, podemos definir las coordenadas en independientes o dependientes. Por coordenadas independientes se entiende aquellas que no dependen la una de la otra. Habrá tantas coordenadas

independientes como grados de libertad tenga nuestro sistema.[6] En el caso de que se esté utilizando un mayor número de coordenadas que grados de libertad, la diferencia entre estos dos números nos dará la cantidad de coordenadas dependientes, así como el número de restricciones necesarias para definir correctamente el sistema.

### 2.3 Problema de posición

El problema de posición trata de averiguar la posición y orientación de todos los sólidos de un sistema multicuerpo conocidos tantos valores como grados de libertad de los que disponga dicho sistema.[3] Es decir, si se divide el problema en coordenadas independientes y dependientes

$$q = \begin{bmatrix} q_{ind} \\ q_{dep} \end{bmatrix}$$

y un sistema dispone de  $n$  grados de libertad y  $m$  ecuaciones de restricción, el número de coordenadas independientes viene determinado por :

$$g = n - m$$

Si se introducen ahora unos valores determinados a las coordenadas independientes y se diferencian las ecuaciones de restricción en dependientes e independientes, se dispone de un sistema de ecuaciones con  $m$  incógnitas y  $m$  ecuaciones. De modo que conocidos los valores de las coordenadas independientes, se pueden conocer los valores de las dependientes y de esta forma, queda resuelto el problema de posición.

En un sistema de ecuaciones lineales,  $\mathbf{Ax} = \mathbf{b}$ , si se dispone del mismo número de incógnitas que de ecuaciones y la matriz de coeficientes  $\mathbf{A}$  no es singular, entonces se obtiene la solución única como  $\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}$ . En un sistema de ecuaciones no lineales, no se puede obtener siguiendo este método y requiere de métodos iterativos como el método de Newton-Raphson.[3]

Este método se basa en la estimación inicial de un valor de la variable independiente, que se aproxime al valor buscado, y se calcula tanto el valor de la función como la derivada en ese punto, de modo que la próxima iteración se obtiene como:

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}$$

Esta fórmula es aplicada hasta que la diferencia entre ambos términos de la resta es muy pequeña. Dependiendo de la estimación inicial puede que el sistema no converja o que el número de iteraciones sea mayor del esperado.[3]

Adaptando la ecuación del método Newton-Raphson a un sistema multicuerpo con  $q_{dep}$  coordenadas dependientes y  $C$  ecuaciones de restricción se llega a la siguiente expresión [3]:

$$q_{i+1}^{dep} = q_i^{dep} - \left. \frac{\partial C}{\partial q^{dep}} \right|_{q_i^{dep}}^{-1} C(q_i^{dep})$$

Iniciándose el método con una iteración inicial de  $q_0^{dep}$ . El jacobiano de las ecuaciones de restricción se suele escribir como  $C_{q_{dep}}$ , de modo que la ecuación pasa a escribirse de la siguiente forma [3]:

$$q_{i+1}^{dep} = q_i^{dep} - C_{q_{dep}}^{-1} \cdot C(q_i^{dep})$$

## 2.4 Simulación cinemática de sistemas multicuerpo

La simulación cinemática no es más que el problema de posición en un conjunto sucesivo de instantes de tiempo muy cercanos entre sí dentro de un intervalo especificado.[3]

Conocidas las ecuaciones de restricción de un sistema:

$$C(q, t) = 0 \quad \text{siendo } q = [q_{ind} \quad q_{dep}]$$

se pueden calcular las aceleraciones y velocidades en la simulación cinemática de un sistema multicuerpo. Para ello, partimos de las siguientes ecuaciones:

$$\frac{dC(q, t)}{dt} = 0 \quad \frac{d\dot{C}(q, t)}{dt} = 0$$

Que aplicando la regla de la cadena, pasan a ser [3]:

$$\begin{aligned} \dot{C}(q, t) &= \frac{\partial C}{\partial q} \frac{dq}{dt} + \frac{\partial C}{\partial t} = C_q \cdot \dot{q} + C_t = 0 \\ \ddot{C}(q, t) &= C_q \cdot \ddot{q} + \dot{C}_q \cdot \dot{q} + \dot{C}_t = 0 \end{aligned}$$

$C_q$  representa el jacobiano de las ecuaciones de restricción y  $C_t$  la derivada parcial de las ecuaciones de restricción con respecto al tiempo.

Separando estas ecuaciones en independientes y dependientes, llegamos a las siguientes expresiones [3]:

$$\begin{aligned} C_{q_{dep}} \cdot \dot{q}_{dep} + C_{q_{ind}} \cdot \dot{q}_{ind} + C_t &= 0 \quad \rightarrow \quad \dot{q}_{dep} = -C_{q_{dep}}^{-1} (C_{q_{ind}} \cdot \dot{q}_{ind} + C_t) \\ C_{q_{dep}} \cdot \ddot{q}_{dep} + C_{q_{ind}} \cdot \ddot{q}_{ind} + \dot{C}_q \cdot \dot{q} + \dot{C}_t &= 0 \quad \rightarrow \quad \ddot{q}_{dep} = -C_{q_{dep}}^{-1} (C_{q_{ind}} \cdot \ddot{q}_{ind} + \dot{C}_q \cdot \dot{q} + \dot{C}_t) \end{aligned}$$

De modo que conocidos los valores de  $\dot{q}_{ind}$ ,  $\ddot{q}_{ind}$  y los valores de  $C_q$ ,  $C_t$  y sus respectivas derivadas temporales, podemos calcular las velocidades y aceleraciones de las coordenadas dependientes, resolviendo de esta forma el problema de velocidades y de aceleraciones.[3]

## 2.5 Simulación dinámica de sistemas multicuerpo

Las ecuaciones del movimiento de un sistema se pueden obtener aplicando las ecuaciones de Newton-Euler a cada uno de los puntos materiales pertenecientes a dicho sistema. Dado que un sólido rígido se puede considerar como un conjunto de puntos materiales cuyas distancias mutuas se mantienen invariables, el movimiento del centro de masa G de un sólido vendrá dado por la segunda ley de Newton.[4]

$$F^i = m^i a_G^i \quad \text{siendo } F^i = \sum_i F^i \text{ y } a_G^i \text{ la aceleración lineal del centro de masa.}$$

Dicha ecuación se puede descomponer según sus componentes:

$$\begin{bmatrix} F_x^i \\ F_y^i \\ F_z^i \end{bmatrix} = m^i \begin{bmatrix} a_{G_x}^i \\ a_{G_y}^i \\ a_{G_z}^i \end{bmatrix}$$

Además de este término, tenemos que añadir la ecuación de Euler que relaciona los momentos que actúan sobre un sólido con la derivada del momento cinético.[3]

$$\sum_k \bar{M}_k = \frac{d\bar{L}}{dt} + \bar{\omega} \times \bar{L} = \bar{I}_G \bar{\alpha} + \bar{\omega} \times (\bar{I}_G \bar{\omega})$$

De modo que juntando ambas ecuaciones, se obtiene el siguiente sistema [3]:

$$\begin{bmatrix} m^i 1_{3 \times 3} & 0 \\ 0 & \bar{I}^i \end{bmatrix} \begin{bmatrix} a_G^i \\ \bar{\alpha}^i \end{bmatrix} = \begin{bmatrix} F^i \\ \bar{M}^i \end{bmatrix} + \begin{bmatrix} 0 \\ -\bar{\omega}^i \times (\bar{I}^i \bar{\omega}^i) \end{bmatrix}$$

siendo  $F^i$  y  $\bar{M}^i$  el sumatorio de fuerzas y momentos resultantes aplicados sobre el sólido  $i$  respectivamente.

Con el objetivo de escribir las ecuaciones de Newton-Euler en términos de las coordenadas generalizadas y sus derivadas, se utilizan estas ecuaciones cinemáticas [3]:

$$\begin{bmatrix} a_G^i \\ \bar{\alpha}^i \end{bmatrix} = \begin{bmatrix} H^i \\ \bar{G}^i \end{bmatrix} \ddot{q} + \begin{bmatrix} h^i \\ \bar{g}^i \end{bmatrix} \dot{q}$$

$$H^i = \frac{\partial r_i}{\partial q} \quad h^i = \frac{\partial (H^i \dot{q})}{\partial q} \quad \bar{G}^i = \frac{\partial \bar{w}_i}{\partial \dot{q}} \quad \bar{g}^i = \bar{G}^i = \frac{\partial \bar{w}_i}{\partial q}$$

De forma que si definimos las siguientes matrices [3]:

$$\hat{M} = \begin{bmatrix} m^i 1_{3 \times 3} & 0 \\ 0 & \bar{I}^i \end{bmatrix} \quad \hat{Q} = \begin{bmatrix} F^i \\ \bar{M}^i \end{bmatrix} \quad \hat{Q}_v = \begin{bmatrix} 0 \\ -\bar{\omega}^i \times (\bar{I}^i \bar{\omega}^i) \end{bmatrix}$$

$$L = \begin{bmatrix} H^i \\ \bar{G}^i \end{bmatrix} \quad l = \begin{bmatrix} h^i \\ \bar{g}^i \end{bmatrix}$$

El sistema pasa a ser de la siguiente forma [3]:

$$\hat{M}(L\ddot{q} + l\dot{q}) = \hat{Q} + \hat{Q}_v \rightarrow M\ddot{q} = Q + Q_v$$

$$M = L^T \hat{M} L \quad Q = L^T \hat{Q} \quad Q_v = L^T (\hat{Q}_v - \hat{M} l \dot{q})$$

donde  $Q$  esta formado por las fuerzas de reacción asociadas a las restricciones y las fuerzas aplicadas sobre los sólidos.



$$Q = Q_{apl} + Q_{reac}$$

Si se hace uso de los multiplicadores de Lagrange para el cálculo de las fuerzas de reacción generalizadas, nuestro sistema pasa a ser el siguiente [3]:

$$M\ddot{q} = Q_{apl} + Q_v + D^T \lambda \quad Q_{reac} = -D^T \lambda \quad D = \begin{bmatrix} C_q \\ B \end{bmatrix}$$

donde  $C_q$  es el jacobiano de las restricciones holónomas y  $B$  el jacobiano de las restricciones no holónomas.

$$\left. \begin{array}{l} C(q, t) = 0 \\ B(q)\dot{q} = 0 \end{array} \right\} \rightarrow \begin{bmatrix} C_q \\ B \end{bmatrix} \dot{q} + \begin{bmatrix} C_t \\ 0 \end{bmatrix} = 0 = D\dot{q} + E = 0$$

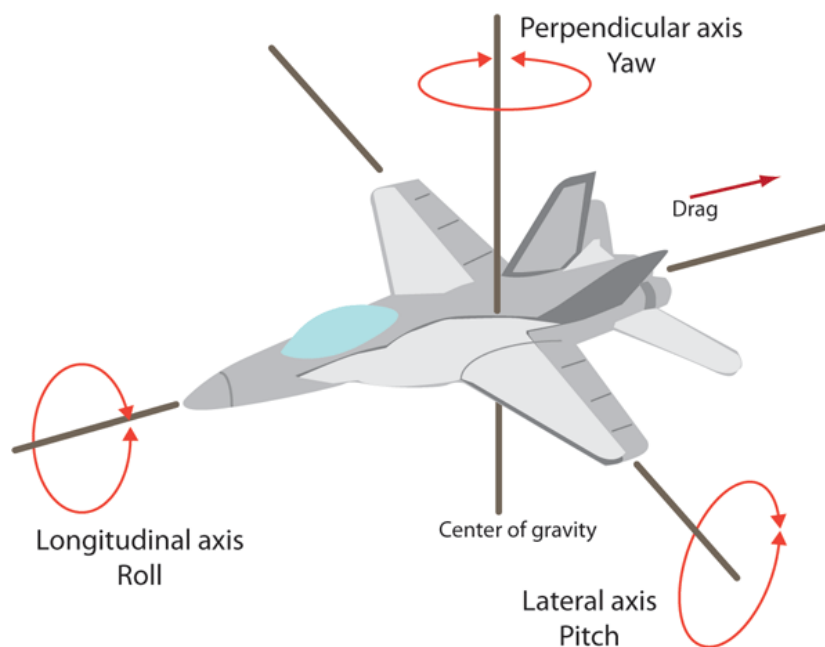
De forma que las ecuaciones del movimiento del sistema son [3]:

$$\begin{array}{l} M\ddot{q} + D^T \lambda = Q_{apl} + Q_v \\ D\ddot{q} + \dot{D}\dot{q} + \dot{E} = 0 \end{array} \rightarrow \begin{bmatrix} M & D^T \\ D & 0 \end{bmatrix} \begin{bmatrix} \ddot{q} \\ \lambda \end{bmatrix} = \begin{bmatrix} Q_{apl} + Q_v \\ -\dot{D}\dot{q} - \dot{E} \end{bmatrix}$$

El problema de posición nos sirve como punto de introducción para la resolución de la simulación cinemática y dinámica de sistemas multicuerpos. Además, el conjunto de ecuaciones descritas en este capítulo nos sirven de punto de partida para el capítulo siguiente en el que se describen estas ecuaciones pero referidas a nuestro problema.

### 3 CINEMÁTICA Y DINÁMICA DEL VEHÍCULO

El vehículo utilizado en todos los casos es el mismo, las variaciones que van a existir dependen del terreno que el vehículo va a seguir y las libertades que se les haya otorgado a dicho vehículo. Para ello, conviene definir a priori los tres ángulos principales de un vehículo conocidos como los ángulos de Euler.



**Figura 1.** Ángulos de Euler. Extraído de <http://www.machinedesign.com/sites/machinedesign.com/files/uploads/2014/06/PRY.gif> > Machinedesign.com </a>.

- Ángulo de guiñada (“Yaw”):  $\psi$ .
- Ángulo de cabeceo (“Pitch”):  $\theta$ .
- Ángulo de balanceo (“Roll”):  $\varphi$ .

Se clasifican los distintos modelos en base a estos ángulos.

- Modelo completo. Los tres ángulos permitidos.
- Modelo simplificado. Nos deshacemos del ángulo de balanceo, de forma que el vehículo exclusivamente puede circular en línea recta por una pendiente y en cualquier dirección sobre el plano.

### 3.1 Modelo completo

El rover está formado por 5 sólidos. Son los siguientes:

- Sólido 2. Chasis.
- Sólido 3. Rueda izquierda.
- Sólido 4. Rueda derecha.
- Sólido 5. Unión chasis sólido 6.
- Sólido 6. Rueda trasera.

Todos los sólidos son rígidos con uniones de par de revolución en las tres ruedas y la unión chasis sólido 6. De las tres ruedas, las dos delanteras son las motorizadas y siempre van a ser paralelas la una a la otra, de forma que para que el rover gire en un sentido requiere de una velocidad de giro mayor de una de las ruedas.

#### 3.1.1 Selección de coordenadas

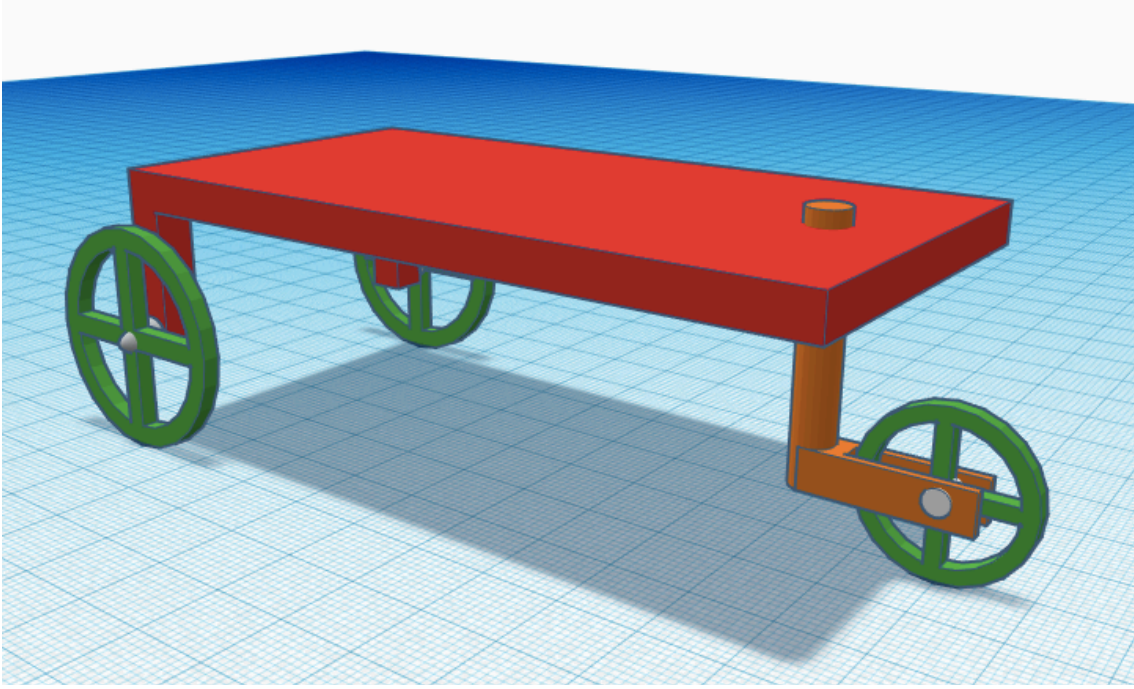
El primer paso para la simulación cinemática y dinámica es la selección de un conjunto adecuado de coordenadas. Para este sistema hemos seleccionado las siguientes:

- Sólido 2. Chasis.
  - Posición del centro de gravedad  $x_2 y_2 z_2$  respecto al sistema de coordenadas global XYZ.
  - Orientación con los ángulos de Euler  $\psi_2 \theta_2 \varphi_2$ .
- Sólido 3. Rueda izquierda.
  - Giro  $\theta_3$ .
- Sólido 4. Rueda derecha.
  - Giro  $\theta_4$ .
- Sólido 5. Unión chasis rueda trasera.
  - Giro  $\psi_5$ .
- Sólido 6. Rueda trasera.
  - Giro  $\theta_6$ .

Además de estas coordenadas, el contacto rígido entre las ruedas y la superficie de la carretera nos obliga a utilizar dos parámetros superficiales por sólido y por contacto, es decir, cuatro coordenadas por cada uno de los puntos de contactos del rover. Sin embargo, hemos supuesto que el grosor de la rueda es mínimo, reduciendo a tres el número de coordenadas necesarias por punto de contacto.

- Punto P. Contacto rueda izquierda (sólido 3).
  - Parámetro de superficie  $s_3$  para la rueda.
  - Parámetros de superficie  $scp = [scp_x \quad scp_y]$  para la carretera.
- Punto Q. Contacto rueda derecha (sólido 4).
  - Parámetro de superficie  $s_4$  para la rueda.
  - Parámetros de superficie  $scq = [scq_x \quad scq_y]$  para la carretera.
- Punto Q. Contacto rueda derecha (sólido 6).
  - Parámetro de superficie  $s_6$  para la rueda.

- Parámetros de superficie  $scs = [scs_x \quad scs_y]$  para la carretera.



**Figura 2.** Modelo Rover en 3D.

De forma, que en total se dispone de 19 coordenadas:

$$q = [x_2 \ y_2 \ z_2 \ \psi_2 \ \theta_2 \ \varphi_2 \ \theta_3 \ s_3 \ \theta_4 \ s_4 \ \psi_5 \ \theta_6 \ s_6 \ scp_x \ scp_y \ scq_x \ scq_y \ scs_x \ scs_y]$$

Estas 19 coordenadas las podemos dividir en dependientes e independientes.

$$q_{dep} = [x_2 \ y_2 \ z_2 \ \psi_2 \ \theta_2 \ \varphi_2 \ s_3 \ s_4 \ \psi_5 \ \theta_6 \ s_6 \ scp_x \ scp_y \ scq_x \ scq_y \ scs_x \ scs_y]$$

$$q_{ind} = [\theta_3 \ \theta_4]$$

De modo que para que el sistema sea compatible necesitamos 17 ecuaciones que serán explicadas con posterioridad en el apartado 3.1.4.

### 3.1.2 Matrices de orientación

Son matrices de orientación aquellas que permiten obtener un vector en un sistema de referencia a partir de ese mismo vector en otro sistema, es decir, permite cambiar un vector de un sistema de referencia a otro.[2] Dado que nuestro sistema es en tres dimensiones, las matrices van a ser de 3x3.

Empezando por el sólido 2, tenemos las siguientes matrices de orientación:

$$A_{\psi_2} = \begin{bmatrix} \cos \psi_2 & -\sin \psi_2 & 0 \\ \sin \psi_2 & \cos \psi_2 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$A_{\psi_2}$  con respecto al sistema de coordenadas global XYZ

$$A_{\theta_2} = \begin{bmatrix} \cos \theta_2 & 0 & -\sin \theta_2 \\ 0 & 1 & 0 \\ \sin \theta_2 & 0 & \cos \theta_2 \end{bmatrix}$$

$A_{\theta_2}$  con respecto al sistema de coordenadas  $x_{\psi_2}y_{\psi_2}z_{\psi_2}$  que surge de la matriz anterior.

$$A_{\varphi_2} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \varphi_2 & \sin \varphi_2 \\ 0 & -\sin \varphi_2 & \cos \varphi_2 \end{bmatrix}$$

$A_{\varphi_2}$  con respecto al sistema de coordenadas  $x_{\theta_2}y_{\theta_2}z_{\theta_2}$  que surge de la matriz anterior.

De modo que la matriz de orientación del sólido 2 es la siguiente:

$A_2 = A_{\psi_2} \cdot A_{\theta_2} \cdot A_{\varphi_2}$  generando un sistema de coordenadas  $x_2y_2z_2$ .

Las matrices de orientación de las ruedas delanteras (sólidos 3 y 4) son las siguientes:

$$A_{\theta_3} = \begin{bmatrix} \cos \theta_3 & 0 & \sin \theta_3 \\ 0 & 1 & 0 \\ -\sin \theta_3 & 0 & \cos \theta_3 \end{bmatrix} \quad A_{\theta_4} = \begin{bmatrix} \cos \theta_4 & 0 & \sin \theta_4 \\ 0 & 1 & 0 \\ -\sin \theta_4 & 0 & \cos \theta_4 \end{bmatrix}$$

En este caso, ambas matrices de orientación son respecto al sistema de coordenadas del chasis  $x_2y_2z_2$ , de forma que obtenemos las matrices con respecto al sistema global multiplicándolas por  $A_2$ .

$$A_3 = A_2 \cdot A_{\theta_3}$$

$$A_4 = A_2 \cdot A_{\theta_4}$$

Las matrices de orientación de los sólidos 5 y 6 son:

$$A_{\psi_5} = \begin{bmatrix} \cos \psi_5 & -\sin \psi_5 & 0 \\ \sin \psi_5 & \cos \psi_5 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad A_{\theta_6} = \begin{bmatrix} \cos \theta_6 & 0 & \sin \theta_6 \\ 0 & 1 & 0 \\ -\sin \theta_6 & 0 & \cos \theta_6 \end{bmatrix}$$

La matriz  $A_{\psi_5}$  es respecto al sistema de coordenadas del chasis, sin embargo, la matriz  $A_{\theta_6}$  es respecto al sistema de coordenadas del sólido 5, de forma que las matrices de orientación respecto al sistema global son las siguientes:

$$A_5 = A_2 \cdot A_{\psi_5}$$

$$A_6 = A_5 \cdot A_{\theta_6}$$

Conocidas las matrices de orientación de todos los sólidos, se puede trabajar en coordenadas locales y posteriormente transformadas a coordenadas globales. El valor unidad en las matrices nos indica con respecto a que eje es el giro, por ejemplo, la matriz  $A_{\psi_5}$  es un giro sobre el eje Z. Las columnas de las matrices ( $A_2 A_3$

$A_4, A_5, A_6$ ) representan los componentes unitarios  $\vec{i}, \vec{j}, \vec{k}$  locales con respecto al sistema de referencia global XYZ.

### 3.1.3 Posición y velocidades de los sólidos

La posición del centro de gravedad del chasis (sólido 2) viene determinada por:

$$r_2 = [x_2 \quad y_2 \quad z_2]$$

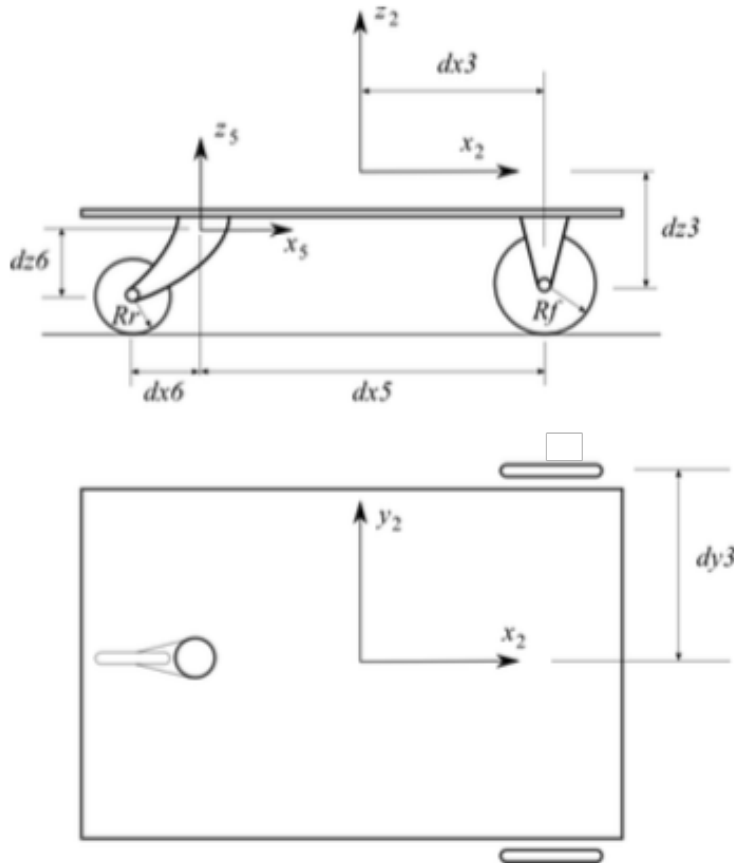
La posición de chasis del rover nos sirve de referencia para calcular la posición del resto de los sólidos. Para ello, tenemos que definir los siguientes vectores locales:

$$\bar{u}_{G_3} = [dx_3 \quad dy_3 \quad -dz_3]$$

$$\bar{u}_{G_4} = [dx_3 \quad -dy_3 \quad -dz_3]$$

$$\bar{u}_{G_5} = [dx_3 - dx_5 \quad 0 \quad (R_f + dz_3) - (R_r + dz_6)]$$

$$\bar{u}_{G_6} = [-dx_6 \quad 0 \quad -dz_6]$$



**Figura 3.** Dimensiones características del Rover.

Una vez que son conocidos la posición en coordenadas locales de dichos puntos, se obtiene en coordenadas globales:

$$r_3 = r_2 + A_2 \cdot \bar{u}_{G_3}$$

$$r_4 = r_2 + A_2 \cdot \bar{u}_{G_4}$$

$$r_5 = r_2 + A_2 \cdot \bar{u}_{G_5}$$

$$r_6 = r_5 + A_5 \cdot \bar{u}_{G_6}$$

Conocida la posición del centro de gravedad de los sólidos, calculamos sus velocidades a partir de los jacobianos de la posición de dichos puntos.

$$v_2 = \frac{\partial r_2}{\partial q} \frac{dq}{dt} = H_2 \cdot \dot{q} \quad v_3 = \frac{\partial r_3}{\partial q} \frac{dq}{dt} = H_3 \cdot \dot{q} \quad v_4 = \frac{\partial r_4}{\partial q} \frac{dq}{dt} = H_4 \cdot \dot{q}$$

$$v_5 = \frac{\partial r_5}{\partial q} \frac{dq}{dt} = H_5 \cdot \dot{q} \quad v_6 = \frac{\partial r_6}{\partial q} \frac{dq}{dt} = H_6 \cdot \dot{q}$$

Además de estas velocidades, necesitamos calcular las velocidades de rotación de los sólidos. Dichas velocidades se obtienen a partir de las matrices de rotación y sus derivadas temporales. Para ello, se deriva cada matriz de rotación con respecto a todos los ángulos del sistema de coordenadas:

$$\dot{A}_i = \frac{\partial A_i}{\partial \psi_2} \dot{\psi}_2 + \frac{\partial A_i}{\partial \theta_2} \dot{\theta}_2 + \frac{\partial A_i}{\partial \varphi_2} \dot{\varphi}_2 + \frac{\partial A_i}{\partial \theta_3} \dot{\theta}_3 + \frac{\partial A_i}{\partial \theta_4} \dot{\theta}_4 + \frac{\partial A_i}{\partial \psi_5} \dot{\psi}_5 + \frac{\partial A_i}{\partial \theta_6} \dot{\theta}_6$$

Una vez conocidas dichas matrices, se calculan las velocidades angulares en coordenadas locales:

$$\bar{W}_i = A_i^T \cdot \dot{A}_i$$

La resolución de dicha ecuación proporciona una matriz con las siguientes componentes:

$$\bar{W}_i = \begin{bmatrix} 0 & -\bar{w}_z^i & \bar{w}_y^i \\ \bar{w}_z^i & 0 & -\bar{w}_x^i \\ -\bar{w}_y^i & \bar{w}_x^i & 0 \end{bmatrix} \text{ de forma que el vector } \bar{w}_i = [\bar{W}_i(3,2) \quad \bar{W}_i(1,3) \quad \bar{W}_i(2,1)]$$

$$\bar{W}_2 = A_2^T \cdot \dot{A}_2 \rightarrow \bar{w}_2 = [\bar{W}_2(3,2) \quad \bar{W}_2(1,3) \quad \bar{W}_2(2,1)]$$

$$\bar{W}_3 = A_3^T \cdot \dot{A}_3 \rightarrow \bar{w}_3 = [\bar{W}_3(3,2) \quad \bar{W}_3(1,3) \quad \bar{W}_3(2,1)]$$

$$\bar{W}_4 = A_4^T \cdot \dot{A}_4 \rightarrow \bar{w}_4 = [\bar{W}_4(3,2) \quad \bar{W}_4(1,3) \quad \bar{W}_4(2,1)]$$

$$\bar{W}_5 = A_5^T \cdot \dot{A}_5 \rightarrow \bar{w}_5 = [\bar{W}_5(3,2) \quad \bar{W}_5(1,3) \quad \bar{W}_5(2,1)]$$

$$\bar{W}_6 = A_6^T \cdot \dot{A}_6 \rightarrow \bar{w}_6 = [\bar{W}_6(3,2) \quad \bar{W}_6(1,3) \quad \bar{W}_6(2,1)]$$

### 3.1.4 Ecuaciones de restricción

Tal y como se ha dicho con anterioridad, para hacer que nuestro sistema sea compatible necesitamos 17 ecuaciones, es decir, tantas como coordenadas dependientes dispone nuestro sistema.

Se puede diferenciar dos tipos de ecuaciones: a) rodadura sin deslizamiento y b) del contacto rígido entre las ruedas y la superficie de la carretera.

#### 3.1.4.1 Rodadura sin deslizamiento

Las ecuaciones de rodadura sin deslizamiento son ecuaciones de restricción no holónomas, es decir, que dependen de la velocidad. A estas ecuaciones se les exige que sean integrables con el objetivo de que sea posible hallar la posición.[7]

Estas ecuaciones exigen que las componentes x e y de la velocidad de las ruedas en los puntos de contactos sean nulas en todo momento. La componente z de la velocidad siempre es nula ya que el rover no se despega del suelo. Estas restricciones implican la formulación de 6 ecuaciones:

$$V_P = \begin{bmatrix} V_{P_x} \\ V_{P_y} \end{bmatrix} = 0$$

$$V_Q = \begin{bmatrix} V_{Q_x} \\ V_{Q_y} \end{bmatrix} = 0$$

$$V_S = \begin{bmatrix} V_{S_x} \\ V_{S_y} \end{bmatrix} = 0$$

Sin embargo, las velocidades  $V_{p_y}$  y  $V_{q_y}$  son las mismas debido al paralelismo existente entre ambas ruedas, de modo que se tiene una ecuación redundante, quedando en este caso restringida a 5 ecuaciones debido a la rodadura sin deslizamiento.

Las velocidades de dichos puntos se obtienen a partir de las velocidades de los centros de gravedad de los sólidos, las posiciones en coordenadas locales de los puntos de contacto y las velocidades angulares de los sólidos.

$$\bar{u}_P = R_f [\sin(\theta_3) \quad 0 \quad -\cos(\theta_3)]$$

$$\bar{u}_Q = R_f [\sin(\theta_3) \quad 0 \quad -\cos(\theta_3)]$$

$$\bar{u}_S = R_r [\sin(\theta_3) \quad 0 \quad -\cos(\theta_3)]$$

Una vez conocida la posición de estos puntos se calcula la velocidad de los puntos de contacto mediante las siguientes ecuaciones:

$$v_P = v_3 + A_3(\bar{w}_3 \times \bar{u}_P) = 0$$



$$v_Q = v_4 + A_4(\bar{w}_4 \times \bar{u}_Q) = 0$$

$$v_S = v_6 + A_6(\bar{w}_6 \times \bar{u}_S) = 0$$

Además, debido a que el rover puede tener las ruedas delanteras en una pendiente, y la rueda trasera en otra pendiente, es necesario proyectar las velocidades con respecto a la tangentes en direcciones x e y de la superficie por la que el rover circula.

$$V_{P_x} = V_P \cdot \frac{tc_{P_x}}{|tc_{P_x}|} \quad V_{P_y} = V_P \cdot \frac{tc_{P_y}}{|tc_{P_y}|}$$

$$V_{Q_x} = V_Q \cdot \frac{tc_{Q_x}}{|tc_{Q_x}|} \quad V_{Q_y} = V_Q \cdot \frac{tc_{Q_y}}{|tc_{Q_y}|}$$

$$V_{S_x} = V_S \cdot \frac{tc_{S_x}}{|tc_{S_x}|} \quad V_{S_y} = V_S \cdot \frac{tc_{S_y}}{|tc_{S_y}|}$$

Los cálculos de las tangentes sobre la superficie serán explicados en el apartado siguiente.

### 3.1.4.2 Contacto rígido

El primer paso a realizar es definir dos parámetros superficiales para cada uno de los sólidos en cada punto de contacto, es decir, necesitamos 12 parámetros superficiales.[8] Sin embargo, dado que se ha supuesto que el grosor de las ruedas es despreciable, solo se necesita un parámetro superficial en el caso de las ruedas, de modo que tenemos 9 parámetros superficiales.

$$S = [s_3 \ s_{cp_x} \ s_{cp_y} \ s_4 \ s_{cq_x} \ s_{cq_y} \ s_6 \ s_{cs_x} \ s_{cs_y}]$$

Estos parámetros se consideran coordenadas no generalizadas ya que no tienen ni inercias ni fuerzas externas asociadas a ellas [8]. Se obtendrán resolviendo las ecuaciones diferenciales no lineales de restricción.

Una vez definidos los parámetros superficiales, se necesita parametrizar las superficies de contacto. Para ello, se utilizan los siguientes vectores de posición en coordenadas locales para las ruedas:

$$\bar{u}_{P_3} = R_f[\sin(s_3) \ 0 \ -\cos(s_3)]$$

$$\bar{u}_{Q_4} = R_f[\sin(s_4) \ 0 \ -\cos(s_4)]$$

$$\bar{u}_{S_6} = R_f[\sin(s_6) \ 0 \ -\cos(s_6)]$$

y estos vectores para la carretera:

$$\bar{u}_{c_P} = [s_{cp_x} \ s_{cp_y} \ 0]$$

$$\begin{aligned}\bar{u}_{cQ} &= [s_{cq_x} \quad s_{cq_y} \quad 0] \\ \bar{u}_{cS} &= [s_{cs_x} \quad s_{cs_y} \quad 0]\end{aligned}$$

Además, se necesitan las tangentes en el punto de contacto, en ambas direcciones, las cuales calculamos como:

$$\begin{aligned}\bar{t}_{P_3} &= \frac{\partial \bar{u}_{P_3}}{\partial s_3} = R_f \begin{bmatrix} \cos(s_3) \\ 0 \\ \sin(s_3) \end{bmatrix} & \bar{t}_{cP_x} &= \frac{\partial \bar{u}_{cP}}{\partial s_{cp_x}} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} & \bar{t}_{cP_y} &= \frac{\partial \bar{u}_{cP}}{\partial s_{cp_y}} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \\ \bar{t}_{Q_4} &= \frac{\partial \bar{u}_{Q_4}}{\partial s_4} = R_f \begin{bmatrix} \cos(s_4) \\ 0 \\ \sin(s_4) \end{bmatrix} & \bar{t}_{cQ_x} &= \frac{\partial \bar{u}_{cQ}}{\partial s_{cq_x}} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} & \bar{t}_{cQ_y} &= \frac{\partial \bar{u}_{cQ}}{\partial s_{cq_y}} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \\ \bar{t}_{S_6} &= \frac{\partial \bar{u}_{S_6}}{\partial s_6} = R_f \begin{bmatrix} \cos(s_6) \\ 0 \\ \sin(s_6) \end{bmatrix} & \bar{t}_{cS_x} &= \frac{\partial \bar{u}_{cS}}{\partial s_{cs_x}} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} & \bar{t}_{cS_y} &= \frac{\partial \bar{u}_{cS}}{\partial s_{cs_y}} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}\end{aligned}$$

Las normales en los puntos se calculan con el producto escalar de las dos componentes tangenciales.

En cuanto a las ecuaciones de restricción, se pueden distinguir dos tipos:

- Los puntos de contacto coinciden, es decir, la posición en coordenadas globales usando estos parámetros de un sólido respecto al otro debe ser la misma.[8]
- Las normales de las dos superficies en el punto de contacto deben ser paralelas.[8]

Estas condiciones implican que necesitamos tener tanto la posición de los puntos de contacto, como las tangentes en coordenadas globales.

$$\begin{aligned}r_{P_3} &= r_3 + A_3 \bar{u}_{P_3} & r_{cP} &= r_7 + A_7 \bar{u}_{cP} \\ r_{Q_4} &= r_4 + A_4 \bar{u}_{Q_4} & r_{cQ} &= r_7 + A_7 \bar{u}_{cQ} \\ r_{S_6} &= r_6 + A_6 \bar{u}_{S_6} & r_{cS} &= r_7 + A_7 \bar{u}_{cS}\end{aligned}$$

Siendo  $r_7$  la posición en globales del origen de la carretera y  $A_7$  la matriz de orientación de la carretera. A su vez, las tangentes en coordenadas globales:

$$\begin{aligned}t_{P_3} &= A_3 \bar{t}_{P_3} & t_{cP_x} &= A_7 \bar{t}_{cP_x} & t_{cP_y} &= A_7 \bar{t}_{cP_y} & n_{cP} &= t_{cP_x} \times t_{cP_y} \\ t_{Q_4} &= A_4 \bar{t}_{Q_4} & t_{cQ_x} &= A_7 \bar{t}_{cQ_x} & t_{cQ_y} &= A_7 \bar{t}_{cQ_y} & n_{cQ} &= t_{cQ_x} \times t_{cQ_y} \\ t_{S_6} &= A_6 \bar{t}_{S_6} & t_{cS_x} &= A_7 \bar{t}_{cS_x} & t_{cS_y} &= A_7 \bar{t}_{cS_y} & n_{cS} &= t_{cS_x} \times t_{cS_y}\end{aligned}$$

De modo, que conocidos todos estos valores, se pueden plantear las ecuaciones de restricción:

- Punto P:  
○  $r_{P_3} - r_{cP} = 0$

- $n_{c_P} \cdot t_{P_3} = 0$
- Punto Q:
  - $r_{Q_4} - r_{c_Q} = 0$
  - $n_{c_Q} \cdot t_{Q_4} = 0$
- Punto S:
  - $r_{S_6} - r_{c_S} = 0$
  - $n_{c_S} \cdot t_{S_6} = 0$

Estas ecuaciones implican un total de 12 ecuaciones de restricción.

A continuación se presenta en la tabla I un resumen del sistema completo.

**Tabla 1.** Comparativa coordenadas y ecuaciones en el modelo completo.

	MODELO COMPLETO	TOTAL
Coordenadas	Dependientes $x_2 \ y_2 \ z_2 \ \psi_2 \ \theta_2 \ \varphi_2 \ s_3 \ s_4 \ \psi_5 \ \theta_6 \ s_6 \ scp_x \ scp_y \ scq_x \ scq_y \ scs_x \ scs_y$	19
	Independientes $\theta_3 \ \theta_4$	
Ecuaciones	Rodadura sin deslizamiento: 5 ecuaciones	17
	Contacto rígido ruedas-carretera: 12 ecuaciones	

De forma, que tal y como hemos expresado con anterioridad, para poder resolver el sistema se tienen que tener tantas ecuaciones como coordenadas dependientes, y la diferencia entre coordenadas y ecuaciones, son los grados de libertad del sistema, en este caso dos grados de libertad.

### 3.1.5 Dinámica del vehículo

Conviene comenzar con la definición de las matrices que se van a utilizar para el cálculo de las ecuaciones del movimiento del rover.

$$\hat{M}_{30 \times 30} = \begin{bmatrix} m^i 1_{3 \times 3} & 0 \\ 0 & \bar{I}^i \end{bmatrix} \quad i = 2, 3, 4, 5, 6.$$

$$\hat{Q}_{30 \times 1} = \begin{bmatrix} F^i \\ \bar{M}^i \end{bmatrix} \quad \hat{Q}_{V_{30 \times 1}} = \begin{bmatrix} 0 \\ -\bar{w}^i \times (\bar{I}^i \bar{w}^i) \end{bmatrix}$$

$$L_{30 \times 19} = \begin{bmatrix} H^i \\ \bar{G}^i \end{bmatrix} \quad l_{30 \times 19} = \begin{bmatrix} h^i \\ \bar{g}^i \end{bmatrix}$$

Una vez conocidas estas matrices, se pueden realizar los cálculos necesarios para transformar el sistema de modo que se utilicen coordenadas generalizadas y sus derivadas temporales.

$$M_{19 \times 19} = L_{19 \times 30}^T \hat{M}_{30 \times 30} L_{30 \times 19}$$

$$\begin{aligned}\hat{Q}_{30x1} &= \hat{Q}_{apl_{30x1}} + \hat{Q}_{reac_{30x1}} & Q_{V_{19x1}} &= L_{19x30}^T (\hat{Q}_{V_{30x1}} - \hat{M}_{30x30} l_{30x19} \dot{q}_{19x1}) \\ Q_{apl_{19x1}} &= L_{19x30}^T \hat{Q}_{apl_{30x1}} & \hat{Q}_{reac_{19x1}} &= D_{19x17}^T \lambda_{17x1} & D_{17x19} &= \begin{bmatrix} C_{q_{12x19}} \\ B_{5x19} \end{bmatrix}\end{aligned}$$

Las ecuaciones de movimiento del sistema son las siguientes:

$$\begin{aligned}M_{19x19} \ddot{q}_{19x1} + D_{19x17}^T \lambda_{17x1} &= Q_{apl_{19x1}} + Q_{V_{19x1}} \rightarrow \begin{bmatrix} M & D^T \\ D & 0 \end{bmatrix} \begin{bmatrix} \ddot{q} \\ \lambda \end{bmatrix} = \begin{bmatrix} Q_{apl} + Q_v \\ -\dot{D}\dot{q} \end{bmatrix} \\ D_{17x19} \ddot{q}_{19x1} + \dot{D}_{17x19} \dot{q}_{19x1} &= 0\end{aligned}$$

En este caso, el término debido a la derivada respecto al tiempo de las restricciones holónomas es nulo, es decir,  $E=0$ . Las fuerzas generalizadas aplicadas sobre el sólido son, las gravitatorias y las provocadas por las ruedas motrices.

$$\hat{Q}_{grav_{30x1}} = \begin{bmatrix} -m^i g \\ 0 \end{bmatrix} \text{ siendo } g: \text{aceleración de la gravedad}$$

$$Q_{mot_{19x1}} = G_{3_{19x3}}^T \begin{bmatrix} 0 \\ Par_{M3} \\ 0 \end{bmatrix} + G_{4_{19x3}}^T \begin{bmatrix} 0 \\ Par_{M4} \\ 0 \end{bmatrix}$$

Sin embargo, dado que se desconocen el valor de los pares en las ruedas, se incluyen estos términos como incógnitas adicionales del problema. Además, dividimos el problema en términos dependientes e independientes, dado que ya se conocen los independientes.

$$\begin{aligned}T_{2x1} &= \begin{bmatrix} Par_{M3} \\ Par_{M4} \end{bmatrix} & S_{19x2} &= [G_{3_{19x1}}^T(2) \quad G_{4_{19x1}}^T(2)] & G_{i_{19x1}}^T(2): \text{segunda fila matriz} \\ \begin{bmatrix} M_{dep_{19x17}} & -S_{19x2} & D_{19x17}^T \\ D_{dep_{17x17}} & 0_{17x2} & 0_{17x17} \end{bmatrix} \begin{bmatrix} \ddot{q}_{dep_{17x1}} \\ T_{2x1} \\ \lambda_{17x1} \end{bmatrix} &= \begin{bmatrix} Q_{grav_{19x1}} + Q_{V_{19x1}} - M_{ind_{19x2}} \ddot{q}_{ind_{2x1}} \\ -\dot{D}_{17x19} \dot{q}_{19x1} - D_{ind_{17x2}} \ddot{q}_{ind_{2x1}} \end{bmatrix}\end{aligned}$$

Sistema que podemos convertir en:

$$A_{36x36} = x_{36x1} b_{36x1}$$

Sistema que se podrá resolver como  $x=A \backslash b$  siempre que la matriz A no sea singular.

## 3.2 Modelo simplificado

### 3.2.1 Selección de coordenadas

Dado que se ha eliminado el ángulo de balanceo del rover, debido a las características de la carretera por la que va a circular, las coordenadas del rover pasan a ser las siguientes:

$$q = [x_2 \ y_2 \ z_2 \ \psi_2 \ \theta_2 \ \theta_3 \ s_3 \ \theta_4 \ s_4 \ \psi_5 \ \theta_6 \ s_6 \ scp_x \ scp_y \ scq_x \ scq_y \ scs_x \ scs_y]$$

De modo que en este modelo, disponemos de 18 coordenadas, 2 coordenadas independientes y 16 dependientes.

$$q_{dep} = [x_2 \ y_2 \ z_2 \ \psi_2 \ \theta_2 \ s_3 \ s_4 \ \psi_5 \ \theta_6 \ s_6 \ scp_x \ scp_y \ scq_x \ scq_y \ scs_x \ scs_y]$$

$$q_{ind} = [\theta_3 \ \theta_4]$$

### 3.2.2 Matrices de rotación

Mismas matrices que en el caso del modelo completo, con la variación de la matriz de rotación del chasis (sólido 2) que interfiere en el resto de matrices.

$$A_2 = A_{\psi_2} \cdot A_{\theta_2}$$

$$A_3 = A_2 \cdot A_{\theta_3}$$

$$A_4 = A_2 \cdot A_{\theta_4}$$

$$A_5 = A_2 \cdot A_{\psi_5}$$

$$A_6 = A_5 \cdot A_{\theta_6}$$

### 3.2.3 Posición y velocidades de los sólidos

Las ecuaciones para obtener la posición de todos los sólidos y sus velocidades son las mismas que en el modelo completo, con la única diferencia de que desaparece el ángulo de balanceo, simplificando de esta forma los cálculos.

$$r_2 = [x_2 \ y_2 \ z_2] \quad v_2 = H_2 \cdot \dot{q} \quad \bar{W}_2 = A_2^T \cdot \dot{A}_2 \rightarrow \bar{w}_2 = [\bar{W}_2(3,2) \ \bar{W}_2(1,3) \ \bar{W}_2(2,1)]$$

$$r_3 = r_2 + A_2 \cdot \bar{u}_{G_3} \quad v_3 = H_3 \cdot \dot{q} \quad \bar{W}_3 = A_3^T \cdot \dot{A}_3 \rightarrow \bar{w}_3 = [\bar{W}_3(3,2) \ \bar{W}_3(1,3) \ \bar{W}_3(2,1)]$$

$$r_4 = r_2 + A_2 \cdot \bar{u}_{G_4} \quad v_4 = H_4 \cdot \dot{q} \quad \bar{W}_4 = A_4^T \cdot \dot{A}_4 \rightarrow \bar{w}_4 = [\bar{W}_4(3,2) \ \bar{W}_4(1,3) \ \bar{W}_4(2,1)]$$

$$r_5 = r_2 + A_2 \cdot \bar{u}_{G_5} \quad v_5 = H_5 \cdot \dot{q} \quad \bar{W}_5 = A_5^T \cdot \dot{A}_5 \rightarrow \bar{w}_5 = [\bar{W}_5(3,2) \ \bar{W}_5(1,3) \ \bar{W}_5(2,1)]$$

$$r_6 = r_5 + A_5 \cdot \bar{u}_{G_6} \quad v_6 = H_6 \cdot \dot{q} \quad \bar{W}_6 = A_6^T \cdot \dot{A}_6 \rightarrow \bar{w}_6 = [\bar{W}_6(3,2) \ \bar{W}_6(1,3) \ \bar{W}_6(2,1)]$$

donde  $\dot{A}_i$  han sido calculados con la siguiente ecuación:

$$\dot{A}_i = \frac{\partial A_i}{\partial \psi_2} \dot{\psi}_2 + \frac{\partial A_i}{\partial \theta_2} \dot{\theta}_2 + \frac{\partial A_i}{\partial \theta_3} \dot{\theta}_3 + \frac{\partial A_i}{\partial \theta_4} \dot{\theta}_4 + \frac{\partial A_i}{\partial \psi_5} \dot{\psi}_5 + \frac{\partial A_i}{\partial \theta_6} \dot{\theta}_6$$

### 3.2.4 Ecuaciones de restricción

Igual que en el modelo completo, se proponen ecuaciones por la rodadura sin deslizamiento y por el contacto rígido existente entre las ruedas y la carretera.

#### 3.2.4.1 Rodadura sin deslizamiento

Estas ecuaciones son exactamente igual que en el modelo completo.

$$\begin{aligned} V_{P_x} &= V_P \cdot \frac{tc_{Px}}{|tc_{Px}|} = 0 & V_{P_y} &= V_P \cdot \frac{tc_{Py}}{|tc_{Py}|} = 0 \\ V_{Q_x} &= V_Q \cdot \frac{tc_{Qx}}{|tc_{Qx}|} = 0 \\ V_{S_x} &= V_S \cdot \frac{tc_{Sx}}{|tc_{Sx}|} = 0 & V_{S_y} &= V_S \cdot \frac{tc_{Sy}}{|tc_{Sy}|} = 0 \end{aligned}$$

Como se puede observar, en el modelo simplificado también se mantiene el paralelismo de las ruedas delanteras, de modo que la ecuación de la  $V_{Q_y} = 0$  es redundante.

#### 3.2.4.2 Contacto rígido

En este caso, existe una pequeña variación con respecto al caso del modelo completo, ya que una de las ecuaciones es redundante. Las normales de las ruedas delanteras son las mismas, ya que el radio de ambas ruedas es el mismo, de forma que lo que hace que varíen las ecuaciones son las tangentes en los puntos de contacto P y Q de la carretera.

Dado que la superficie por la que el vehículo va a circular es totalmente plana y va a circular en línea recta, las tangentes en ambos puntos son también las mismas, de modo que ya se conoce la ecuación redundante del sistema. En este caso, las ecuaciones son las siguientes:

- Punto P:
  - $r_{P_3} - r_{c_P} = 0$
  - $n_{c_P} \cdot t_{P_3} = 0$
- Punto Q:
  - $r_{Q_4} - r_{c_Q} = 0$
- Punto S:
  - $r_{S_6} - r_{c_S} = 0$
  - $n_{c_S} \cdot t_{S_6} = 0$

En este caso, disponemos de un total de 11 ecuaciones del contacto rígido, siendo esta la principal diferencia con respecto al caso anterior. Nuestro sistema es compatible, tal y como se puede observar en la siguiente

tabla, dado que se ha eliminado una coordenada y una ecuación, manteniendo los dos grados de libertad del sistema.

**Tabla 2.** Comparativa coordenadas y ecuaciones del modelo simplificado.

	MODELO SIMPLIFICADO	TOTAL
Coordenadas	Dependientes $x_2 \ y_2 \ z_2 \ \psi_2 \ \theta_2 \ s_3 \ s_4 \ \psi_5 \ \theta_6 \ s_6 \ scp_x \ scp_y \ scq_x \ scq_y \ scs_x \ scs_y$	18
	Independientes $\theta_3 \ \theta_4$	
Ecuaciones	Rodadura sin deslizamiento: 5 ecuaciones	16
	Contacto rígido ruedas-carretera: 11 ecuaciones	

### 3.2.5 Dinámica del vehículo

En este caso, las variaciones existentes dentro de las ecuaciones del movimiento están provocadas por la coordenada que se ha eliminado y la ecuación redundante existente en el caso del contacto rígido entre los sólidos.

En este caso, las ecuaciones del movimiento son las siguientes:

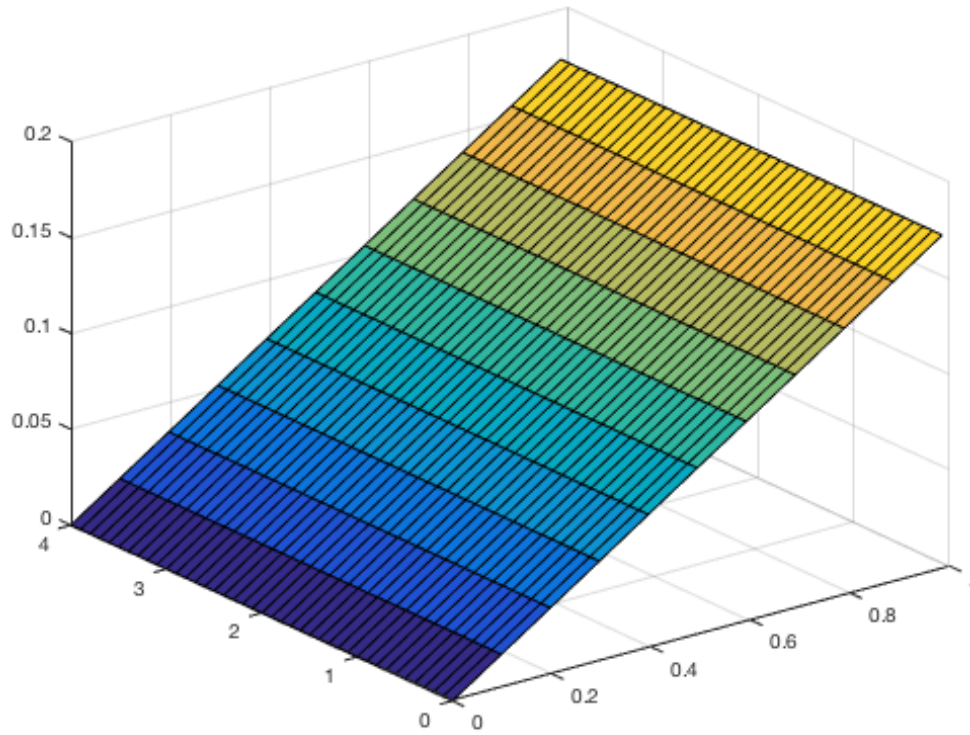
$$\begin{bmatrix} M_{dep_{18 \times 16}} & -S_{18 \times 2} & D_{18 \times 16}^T \\ D_{dep_{16 \times 16}} & 0_{16 \times 2} & 0_{16 \times 16} \end{bmatrix} \begin{bmatrix} \ddot{q}_{dep_{16 \times 1}} \\ T_{2 \times 1} \\ \lambda_{16 \times 1} \end{bmatrix} = \begin{bmatrix} Q_{grav_{18 \times 1}} + Q_{V_{18 \times 1}} - M_{ind_{18 \times 2}} \ddot{q}_{ind_{2 \times 1}} \\ -\dot{D}_{16 \times 18} \dot{q}_{18 \times 1} - D_{ind_{18 \times 2}} \ddot{q}_{ind_{2 \times 1}} \end{bmatrix}$$

$$A_{34 \times 34} x_{34 \times 1} = b_{34 \times 1}$$

Como se puede observar, la diferencia entre las dimensiones del modelo completo y el modelo simplificado coinciden con la reducción de una coordenada generalizada y una ecuación de restricción.

### 3.3 Superficie parametrizada

A la hora de realizar los cálculos necesarios para la simulación dinámica del vehículo, previos a los análisis experimentales, se ha parametrizado la carretera por la que el rover va a circular como un plano inclinado. A pesar de que las dimensiones de estas se suponen infinitas para su cálculo, se ha representado el plano con unas dimensiones acotadas para que fuese posible su representación.



**Figura 4.** Superficie parametrizada

Las dimensiones del plano nos sirven para conocer el valor del ángulo de la pendiente, que es el valor principal con el que se modifican los resultados obtenidos en la simulación. En este caso se ha utilizado un plano con una inclinación de  $10^\circ$ .

### 3.4 Parámetros del vehículo

Dentro de los parámetros del vehículo incluimos las masas de cada uno de los sólidos y los tensores de inercia.

Masa de los sólidos

- $m_2 = 0.7 \text{ kg}$
- $m_3 = 0.15 \text{ kg}$
- $m_4 = 0.15 \text{ kg}$
- $m_5 = 0.2 \text{ kg}$
- $m_6 = 0.1 \text{ kg}$

A pesar de que los valores de los tensores de inercias no son calculados con exactitud, se ha realizado una estimación de ellos, de forma que el mayor de todos sea el del sólido 2 o chasis del Rover.

$$\bar{I}_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \bar{I}_3 = \bar{I}_4 = \begin{bmatrix} 0.01 & 0 & 0 \\ 0 & 0.01 & 0 \\ 0 & 0 & 0.01 \end{bmatrix}$$



$$\bar{I}_5 = \bar{I}_6 = \begin{bmatrix} 0.1 & 0 & 0 \\ 0 & 0.1 & 0 \\ 0 & 0 & 0.1 \end{bmatrix}$$

Este capítulo nos ha proporcionado las ecuaciones que se necesitan para resolver la simulación del Rover, tanto en el modelo simplificado como en el modelo completo, sin embargo, la experimentación con el Rover se va a realizar en el caso completo y el modelo simplificado nos ha servido como base para la realización del programa en el caso completo. En el siguiente capítulo se describen los programas utilizados para la resolución de estas ecuaciones.



# 4 PROGRAMAS UTILIZADOS PARA LA SIMULACIÓN

---

En este capítulo se describen los diferentes tipos de programas utilizados a la hora de realizar la simulación del rover. Se pueden distinguir tres programas principales, uno para el cálculo simbólico de todos los parámetros necesarios, otro para la resolución de las ecuaciones del movimiento del sistema y un tercer programa para la animación del rover. Los programas están incluidos en los anexos.

## 4.1 Cálculo simbólico

En este programa, se ha trabajado con el paquete simbólico de MATLAB que te permite realizar los cálculos sin necesidad de proporcionarle valores numéricos. Para ello, es necesario introducirle al programa todos los parámetros que va a necesitar en simbólico.

Dentro de este programa, se realizan los cálculos de los siguientes valores:

- Matrices de orientación y sus derivadas temporales
- Posición y velocidades de cada uno de los centros de gravedad del sólido.
- Jacobianos del vector posición y de la velocidad angular
- Restricciones del contacto rígido entre sólidos
- Restricciones rodadura sin deslizamiento
- Matriz jacobiano de las restricciones
- Matrices de masa e inercia
- Fuerzas generalizadas cuadráticas en velocidad y gravitatorias
- Matrices de transformación cinemática

Una vez calculados todos estos parámetros, se han realizados funciones específicas que posteriormente serán utilizadas tanto en el programa para la simulación dinámica como en la animación.

Como ejemplo de funciones creadas, la función `MatrizD`, que recibiendo los valores de las coordenadas generalizadas devuelve la matriz jacobiano de las restricciones.

## 4.2 Simulación dinámica inversa

El primer paso que se ha realizado en la simulación dinámica es el cálculo de los valores iniciales de las coordenadas del sistema con el que se inicia el bucle iterativo. El cálculo se ha realizado mediante la función “`fsolve`” que realiza un cálculo similar al método de Newton-Raphson descrito en el apartado 2.3. Para ello, la función “`fsolve`” requiere una función y unos valores estimados de la resolución con los que asegurarse que la solución converja.

La función utilizada ha sido la generada por el conjunto de restricciones holónomas del contacto rígido; la estimación de los valores iniciales se ha realizado mediante trigonometría básica.

Una vez que hemos calculado estos valores, se inicia el bucle para la resolución del sistema, siguiendo los siguientes pasos:

1. Se obtiene la matriz  $D$  (jacobiano de las restricciones), proporcionándole las posiciones ( $q$ ), y se divide en parte dependiente e independiente.
2. Calculamos las velocidades dependientes ( $\dot{q}_{dep}$ ) a partir de la siguiente ecuación:

$$\dot{q}_{dep} = -D_{dep} \setminus (D_{ind} \dot{q}_{ind})$$

3. Se calcula matriz  $\dot{D}$ , introduciendo en la función que la determina, los valores de las posiciones ( $q$ ) y las velocidades ( $\dot{q}$ ).
4. Se obtienen las matrices de masa ( $M$ ), fuerzas generalizadas cuadráticas en velocidad ( $Q_v$ ) y fuerzas generalizadas gravitacionales ( $Q_{grav}$ ) llamando a la función “MatMasaFuerzas” que nos proporciona dichos valores, y dividimos la matriz de masa en parte dependiente e independiente.
5. Creamos la matriz  $A$  y el vector  $b$ , que nos permite resolver el sistema mediante  $x = A \setminus b$ .

$$A = \begin{bmatrix} M_{dep} & -S & D^T \\ D_{dep} & 0 & 0 \end{bmatrix} \quad b = \begin{bmatrix} Q_{grav} + Q_v - M_{ind} \ddot{q}_{ind} \\ -\dot{D} \dot{q} - D_{ind} \ddot{q}_{ind} \end{bmatrix}$$

6. Una vez calculadas las aceleraciones, se integran y se obtienen los valores de las posiciones en el instante especificado.
7. Estos valores calculados pasan a ser los valores iniciales con los que se vuelve a repetir el proceso hasta el último instante de tiempo.
8. Se llama a la función “Anima” que realiza la animación del movimiento del Rover.

### 4.3 Animación

El programa que se encarga de realizar la animación recibe como datos los valores de las coordenadas en posición y el ángulo de la pendiente de la carretera por la que el Rover circula, y realiza una representación en tres dimensiones del movimiento del rover a lo largo del tiempo.

Requiere de un bucle que representa en cada uno de los instantes de tiempo los componentes del rover y la carretera por la que circula.

### 4.4 Registro de Datos

Los datos registrados por los sensores del Rover requieren de ser tratados, tal y como posteriormente explicaremos en el apartado 5.2, por ello se ha requerido crear un programa que recoja estos datos y los trate para su posterior uso en los programas descritos anteriormente.

Inicialmente se requiere la lectura de los datos registrados por el Rover, los cuales son extraídos en un archivo “.txt”. Estos datos se encuentran en sistemas de unidades diferentes a los que necesitamos por lo que el primer paso a realizar es convertir estos datos al sistema que necesitamos. A su vez, debido al proceso de adquisición de datos, los datos registrados hasta el instante  $t=1s$  son incorrectos [7], de forma que se deben desechar.

Una vez registrado los datos de los ángulos recorridos por las ruedas, en el sistema de unidades correcto, se calculan las velocidades y aceleraciones mediante diferencias finitas, y posteriormente se tratan con un filtro

que será explicado en el apartado 5.2. Una vez que disponemos de toda la información necesaria para la simulación, se genera una matriz con estos datos para que posteriormente puedan ser extraídos del programa de simulación dinámica.

## 4.5 Funciones adicionales

Además de los programas principales explicados anteriormente y de las funciones desarrolladas por el programa del cálculo simbólico, se han utilizado varias funciones adicionales con el objetivo de simplificar el método resolutivo.

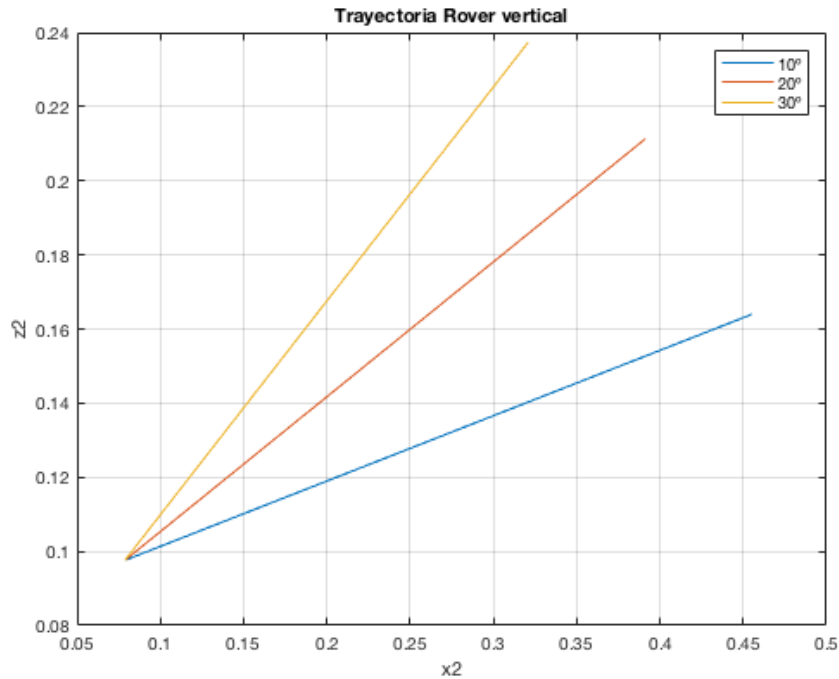
- *ReordenaCoord*: Función que recibe las coordenadas dependientes e independientes de forma separada y las reordena siguiendo el orden que hayamos utilizado a la hora de crear el vector de coordenadas generalizadas  $q$ .
- *ParámetrosRover*: Función que engloba todos los parámetros utilizados. Dimensiones, masas e inercias entre otros.
- *CondicionesIniciales*: Función que calcula las condiciones iniciales de la posición  $(x_2, z_2)$  del chasis del rover mediante trigonometría básica.
- *DerivadaPrimera*: Función que calcula la derivada primera mediante diferencias centradas.
- *DerivadaSegunda*: Función que calcula la derivada segunda mediante diferencias centradas.

## 4.6 Análisis del programa de simulación dinámica

Previos a los análisis experimentales, se han realizado análisis del programa realizado para la simulación dinámica con el objetivo de comprobar que los resultados obtenidos son los esperados, ya que computacionalmente podemos obtener un mayor número de variaciones del problema que experimentalmente. Se han analizado diferentes situaciones y se han representado gráficamente los resultados obtenidos.

### 4.6.1 Modelo simplificado

Dado que no se le ha permitido al vehículo el ángulo de balanceo, la única trayectoria posible para este modelo es en línea recta. En este análisis, se ha realizado el recorrido del vehículo por tres pendientes diferentes, a velocidad constante, y los resultados obtenidos han sido los siguientes:



**Figura 5.** Trayectorias verticales modelo simplificado.

Estos resultados indican que la trayectoria realizada por el rover, medida como la relación existente entre las coordenadas del chasis ( $x_2, z_2$ ), concuerdan con los ángulos introducidos de las pendientes. Los errores obtenidos de los ángulos del rover respecto al ángulo de la carretera han sido los siguientes:

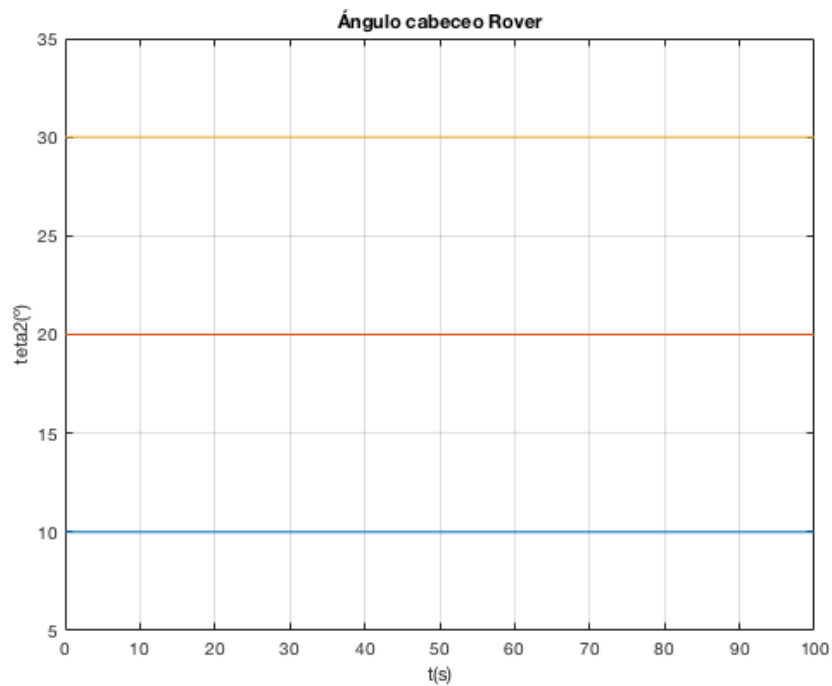
**Tabla 3.** Error ángulo trayectoria vertical respecto ángulo carretera.

Velocidad [rad/s]	Ángulo carretera	Error absoluto
10	10°	$3.7836 \cdot 10^{-12}$
10	20°	$6.029 \cdot 10^{-12}$
10	30°	$5.2224 \cdot 10^{-12}$

El error que se introduce en la simulación se debe al proceso de integración. Este error depende de la velocidad que sea introducida en las ruedas motoras, el paso de tiempo que se utilice en la integración, y las dimensiones del vector tiempo, es decir, la duración de la simulación.

Además de la posición del chasis ( $x_2, z_2$ ), se puede utilizar el ángulo de cabeceo ( $\theta_2$ ) y el recorrido sobre el plano XY como comprobación. En este caso, el recorrido debería de ser una línea recta y este ángulo ( $\theta_2$ ) debería coincidir con el ángulo de la pendiente de la carretera.

Representación gráfica del ángulo de cabeceo de Rover a lo largo de todo el recorrido.



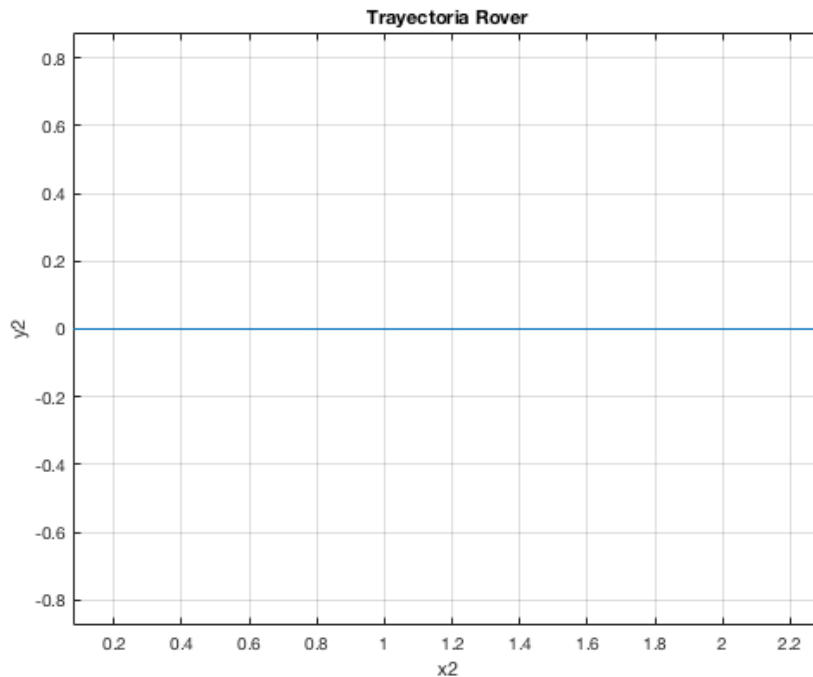
**Figura 6.** Ángulos de cabeceo modelo simplificado.

Se han calculado los errores de éste ángulo con respecto al ángulo de la pendiente, en los tres casos anteriores, y el resultado es el siguiente:

**Tabla 4.** Errores ángulo cabeceo respecto ángulo carretera.

Ángulo carretera	Error absoluto
10°	$5.1366 \cdot 10^{-12}$
20°	$2.8974 \cdot 10^{-11}$
30°	$1.6645 \cdot 10^{-11}$

Representación gráfica de la trayectoria sobre el plano XY global.



**Figura 7.** Trayectorias sobre plano XY modelo simplificado

Como podemos observar, los resultados obtenidos tanto con los ángulos de cabeceo del Rover como con la trayectoria sobre el plano XY coinciden con los esperados, manteniendo un error absoluto del mismo orden que el calculado siguiendo la trayectoria  $(x_2, z_2)$ .

#### 4.6.2 Modelo completo

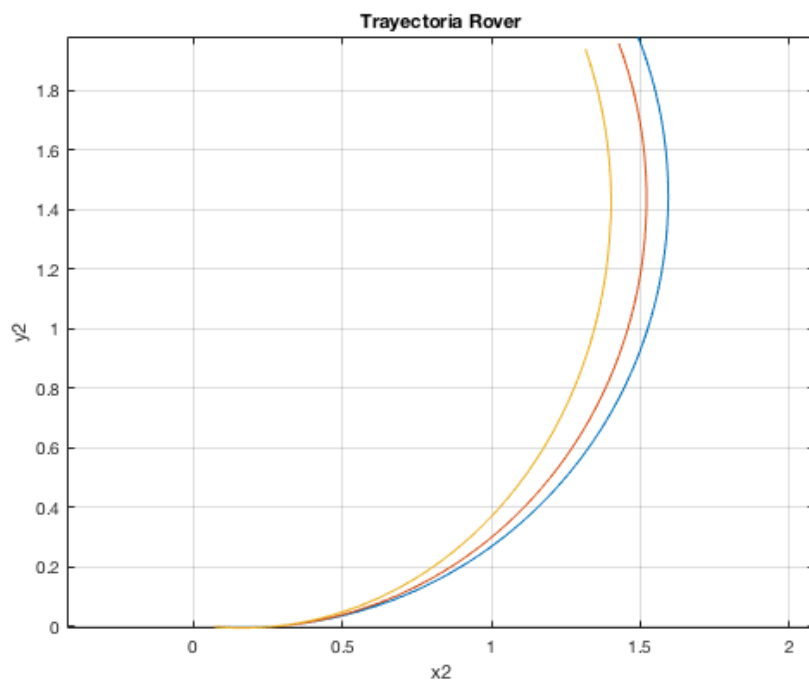
Este modelo permite que el vehículo gire al subir la pendiente ya que le hemos proporcionado el ángulo de balanceo que anteriormente hemos eliminado.

Al igual que en el caso anterior, se ha analizado la trayectoria del vehículo en tres pendientes diferentes, pero con la variación de introducirle velocidades diferentes pero constantes a cada rueda, de forma que debería realizar una trayectoria circular.

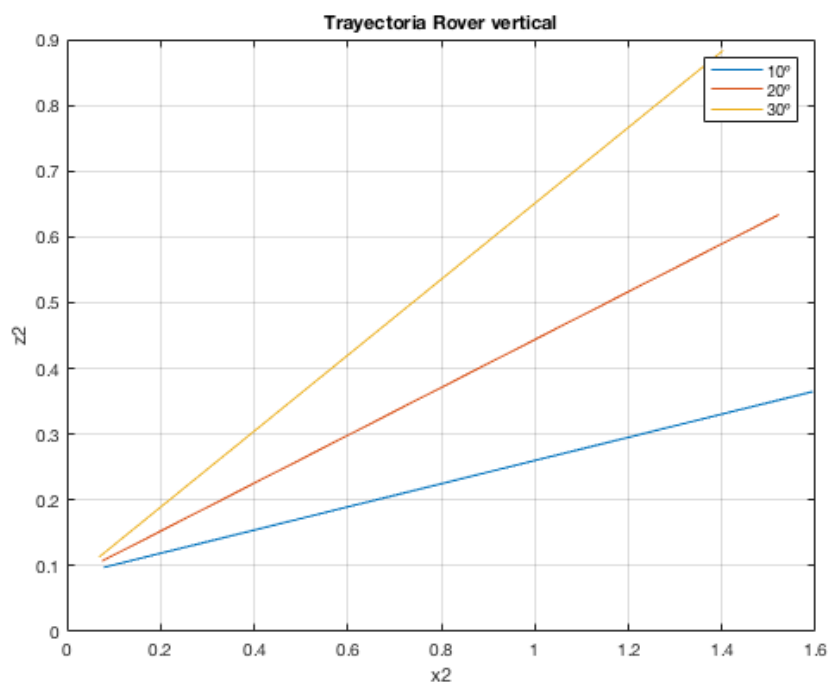
Dentro de éste análisis, se ha representado gráficamente la trayectoria sobre el plano XY global del sistema, la trayectoria seguida por el chasis  $(x_2, z_2)$ , y los respectivos ángulos de cabeceo y balanceo del vehículo.

Los resultados obtenidos son los siguientes:



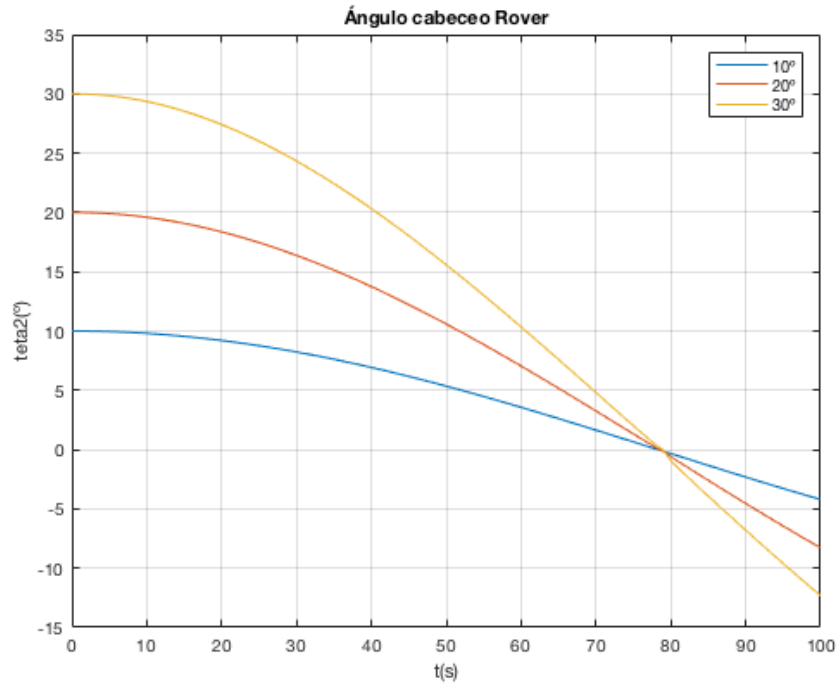


**Figura 8.** Trayectorias sobre plano XY modelo completo.

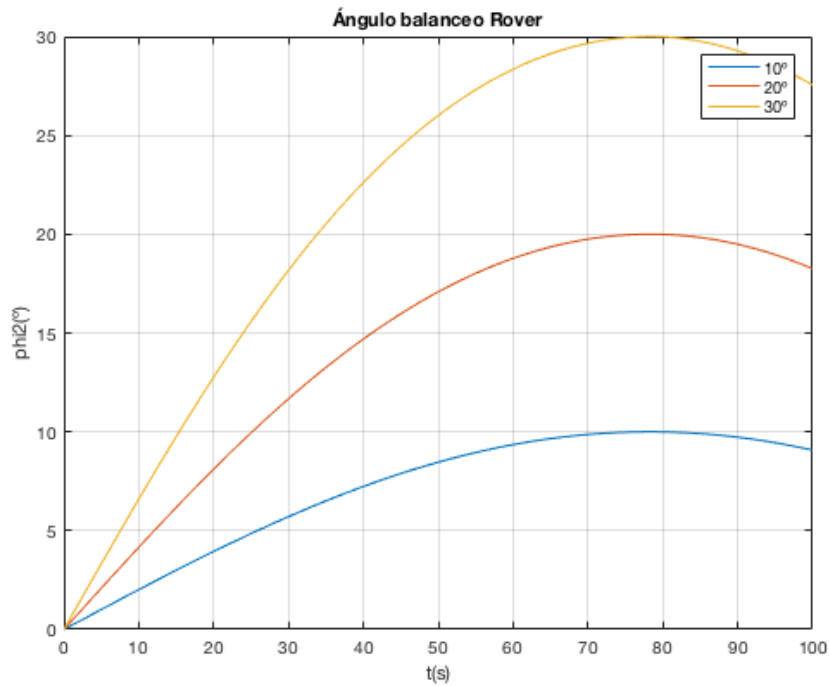


**Figura 9.** Trayectorias verticales modelo completo.

Estas representaciones nos indican que a priori tanto la trayectoria vertical como la trayectoria sobre el plano XY coincide con lo esperado, ya que la velocidad introducida para la rueda derecha era levemente superior a la izquierda, produciendo de esta forma el giro hacia la izquierda deseado.



**Figura 10.** Ángulos de cabeceo modelo completo

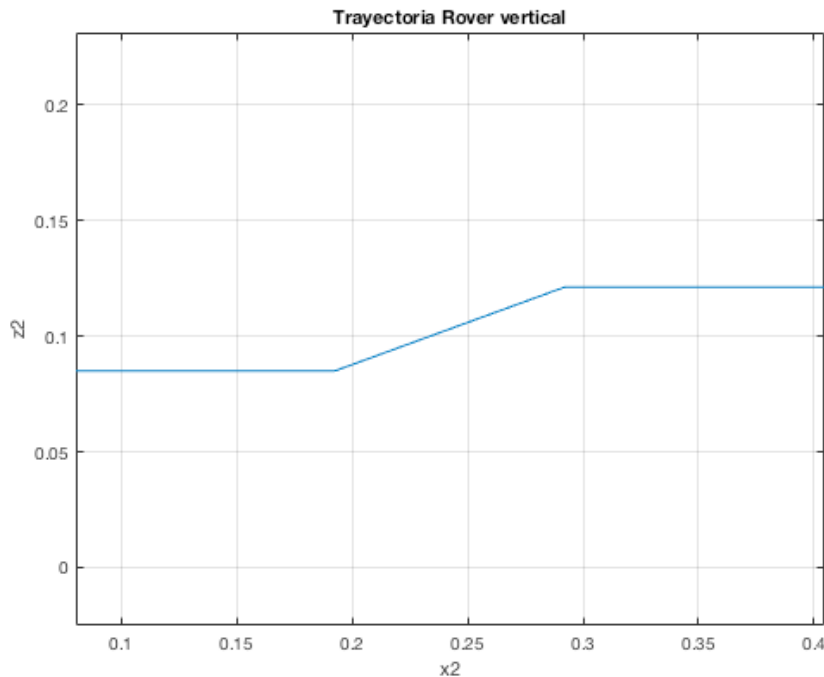


**Figura 11.** Ángulos de balanceo modelo completo

Los resultados obtenidos en estas representaciones también concuerdan con lo esperado, ya que el ángulo de cabeceo inicial coincide con el de la pendiente de la carretera y este disminuye hasta llegar a ser nulo en el punto en el que la trayectoria es totalmente perpendicular a la dirección de subida de la pendiente.

Simultáneamente, el valor del ángulo de balanceo aumenta desde un valor inicial nulo hasta el máximo en el mismo instante en el que el ángulo de cabeceo es nulo, siendo este instante en torno a  $t=80$  s.

Aparte de esta simulación, dado que siempre se ha iniciado la simulación directamente sobre la rampa, hemos decidido dividirlo en tres tramos, los tramos iniciales y finales sin pendiente y el tramo de en medio con una pendiente de  $20^\circ$ . Obteniendo resultados de la figura 12. Estos resultados nos indican que la proyección de las velocidades de los puntos en las tangentes superficiales nos han sido de utilidad para que fuese posible tener las ruedas delanteras en una pendiente y la trasera en otra diferente, de forma que no fuesen paralelos al eje local del chasis.



**Figura 12.** Trayectoria vertical con tres tramos a pendientes diferentes

El objetivo de este apartado era la comprobación con diversas situaciones de que los programas generados nos proporcionaban las soluciones adecuadas, de forma que a expensas de realizar la simulación con los datos experimentales parece que los resultados que obtenemos con los diferentes programas coinciden con lo esperado y podemos pasar al siguiente apartado en el que introducimos como valores de entrada los registrados por los sensores del Rover.

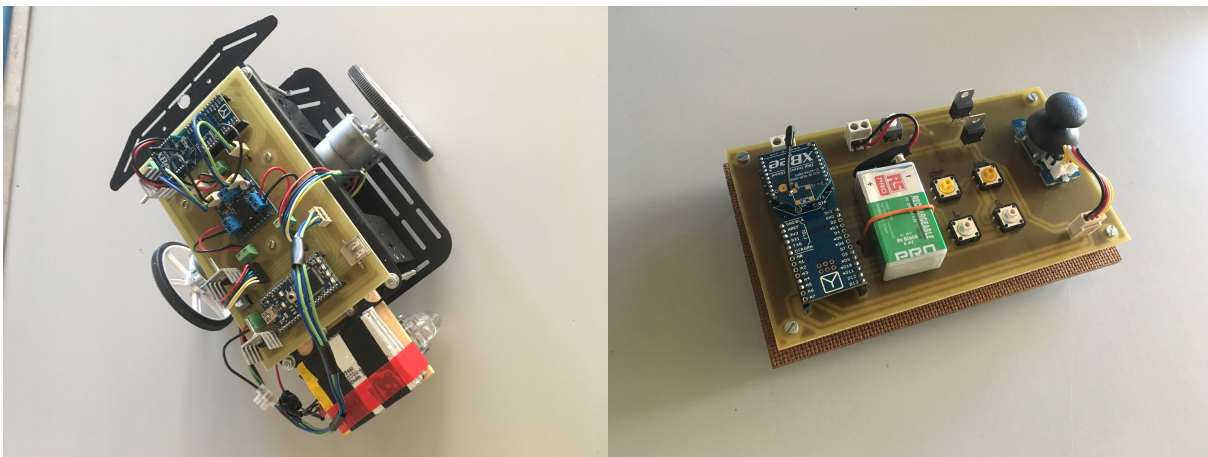
## 5 RESULTADOS EXPERIMENTALES DE LA SIMULACIÓN

Este capítulo está formado por los resultados obtenidos de la simulación que se ha realizado con los datos extraídos directamente del Rover cuando éste realiza un trayecto por una cuesta con cierta pendiente.

### 5.1 Registro de señales digitales

El trayecto realizado por el Rover queda registrado por el conjunto de sensores que éste incluye, siendo estos sensores dos encoders, un giroscopio y un acelerómetro. Obteniendo de esta forma los siguientes datos:

- Columna 1. Instante de tiempo [s].
- Columna 2. Aceleración en dirección X [ $m/s^2$ ].
- Columna 3. Aceleración en dirección Y [ $m/s^2$ ].
- Columna 4. Aceleración en dirección Z [ $m/s^2$ ].
- Columna 5. Velocidad angular en dirección X [grados/s].
- Columna 6. Velocidad angular en dirección Y [grados/s].
- Columna 7. Velocidad angular en dirección Z [grados/s].
- Columna 8. Pulsos de encoder rueda derecha con signo negativo.
- Columna 9. Pulsos de encoder rueda izquierda.



**Figura 13.** Rover y control remoto.

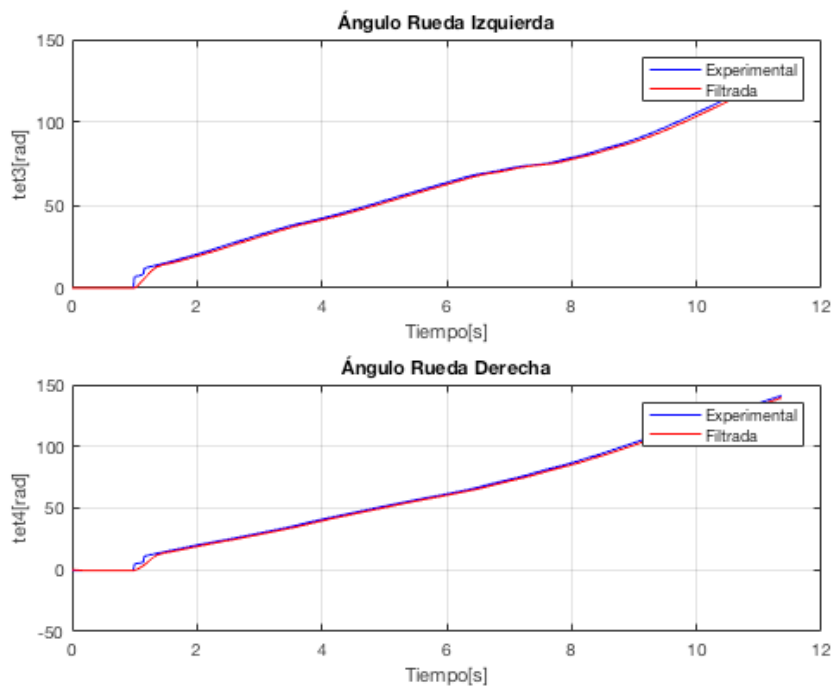
## 5.2 Tratamiento de señales digitales

Los encoders tienen una resolución de 960 pulsos por revolución, de modo que el paso inicial es convertir los pulsos de los encoders en radianes.[7] Una vez conocidos los ángulos girados por ambas ruedas, en radianes, hay que obtener las velocidades y aceleraciones angulares de las ruedas. Estos resultados se han obtenidos mediante el método diferencial de diferencias finitas centradas.

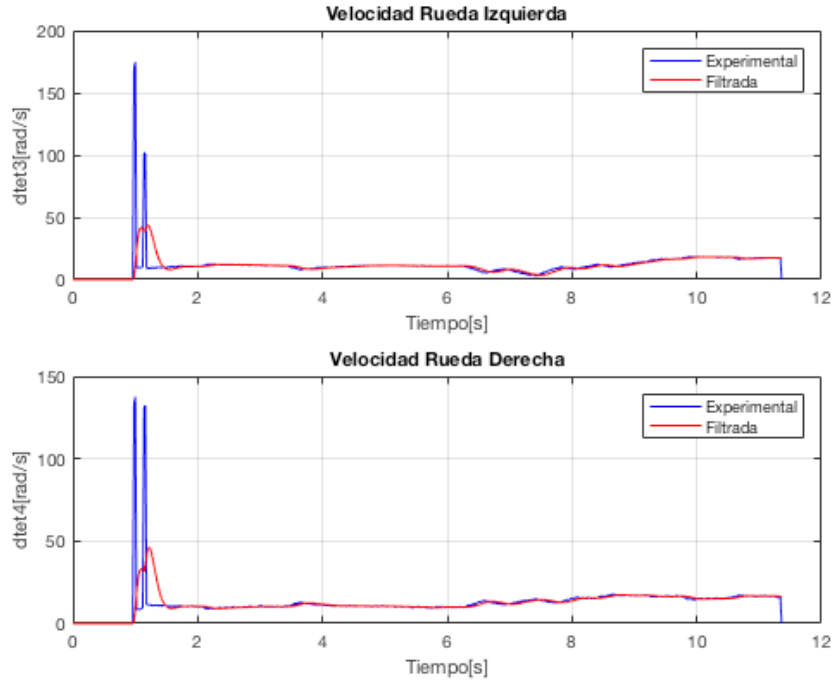
En cuanto a las señales registradas, con el objetivo de conocer digitalmente el espectro de una señal hasta una determinada frecuencia máxima ( $f_{max}$ ), se requiere realizar el muestreo a una frecuencia ( $f_m$ ) al menos el doble de la frecuencia máxima, dado que sino no se podrían conocer los armónicos superiores a la mitad de la frecuencia de muestreo. Además, si existen frecuencias iguales o superiores a  $f_m/2$ , estas frecuencias perturbarían el resultado obtenido. Esta condición es conocida con el nombre de criterio de Nyquist. [9]

Como no es siempre realizar el muestreo a la frecuencia anterior, la señal debe filtrarse con el objetivo de eliminar todas las frecuencias superiores a la de Nyquist, es decir, la mitad de la frecuencia de muestreo. Estos filtros son los denominados filtros de paso bajo, que eliminan todas aquellas frecuencias superiores a la de Nyquist y se permite el paso en frecuencias inferiores, garantizando de esta forma que el espectro obtenido es suficientemente aproximado hasta esa frecuencia.[9] Los filtros se caracterizan por su función de transferencia, de forma que el tipo de filtro dependerá de dicha función.

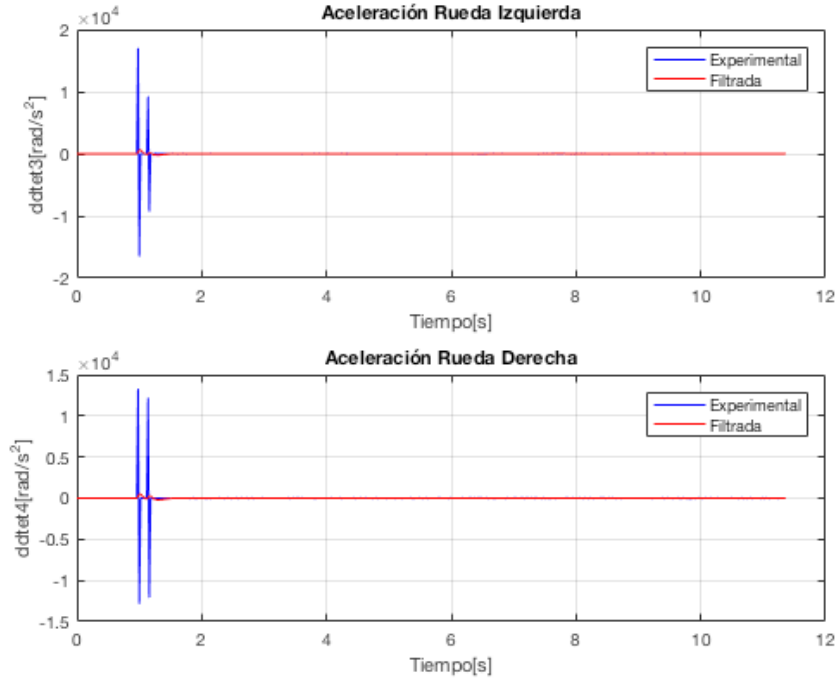
En este caso, se han filtrado las señales de los ángulos, velocidades y aceleraciones angulares de las ruedas con un filtro de butterworth de segundo orden a una frecuencia de corte de 4 Hz.[7] Dicho filtro se ha realizado mediante la función de matlab “butter(N,Wn,'low')” siendo N el orden del filtro y Wn la frecuencia de corte.



**Figura 14.** Ángulos girados por las ruedas



**Figura 15.** Velocidades de las ruedas



**Figura 16.** Aceleraciones de las ruedas

Estas señales son los datos de entrada para la resolución del sistema.

$$q_{ind} = \begin{bmatrix} \theta_3 \\ \theta_4 \end{bmatrix} \quad \dot{q}_{ind} = \begin{bmatrix} \dot{\theta}_3 \\ \dot{\theta}_4 \end{bmatrix} \quad \ddot{q}_{ind} = \begin{bmatrix} \ddot{\theta}_3 \\ \ddot{\theta}_4 \end{bmatrix}$$

### 5.3 Superficie parametrizada

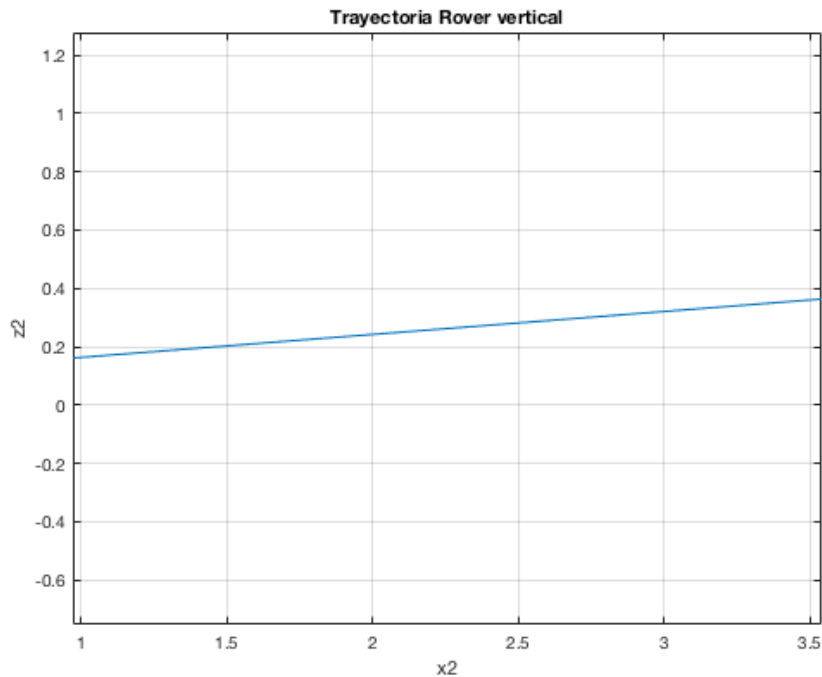
La superficie utilizada para la trayectoria recorrida por el Rover es una de las rampas de la escuela. Para su simulación, necesitamos conocer el valor de su pendiente, por ello, hemos tomado la posición en la dirección X y Z de tres puntos de la rampa y la pendiente obtenida ha sido de  $4.5^\circ$ .



**Figura 17.** Rampa utilizada para la simulación

### 5.4 Resultados

Al igual que en los resultados analíticos obtenidos en el apartado 4.5, se han realizado representaciones de una serie de valores que nos proporcionan la información necesaria para poder confirmar que los resultados analíticos obtenidos con los datos registrados experimentalmente coinciden con lo esperado. Por ello, como punto de partida, se ha representado la trayectoria vertical seguida por el Rover (Figura 16).



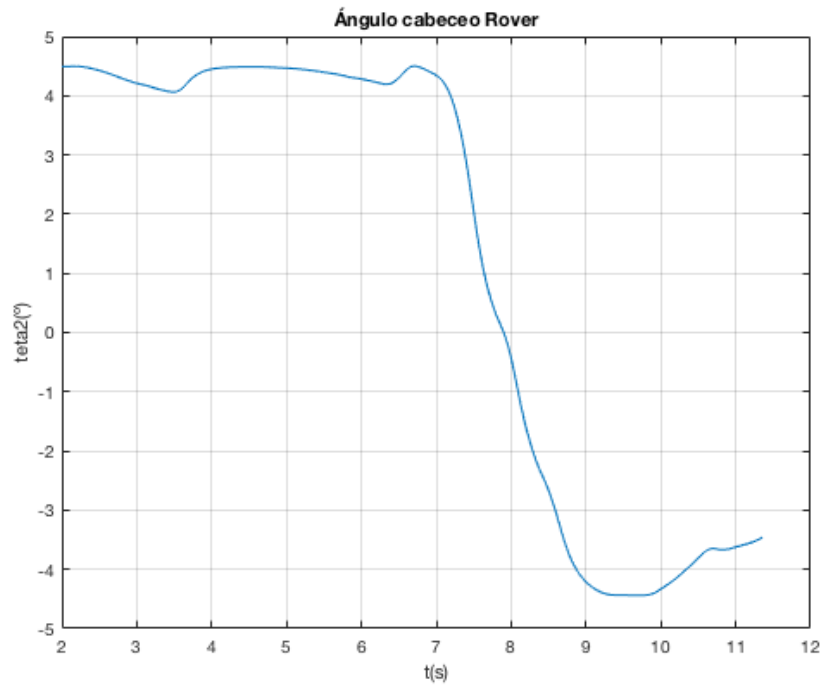
**Figura 18.** Trayectoria vertical experimental

Calculando el ángulo como la diferencia entre el punto inicial y el punto final se obtiene que el ángulo de la trayectoria seguida por el Rover coincide con la del plano utilizado como referencia.

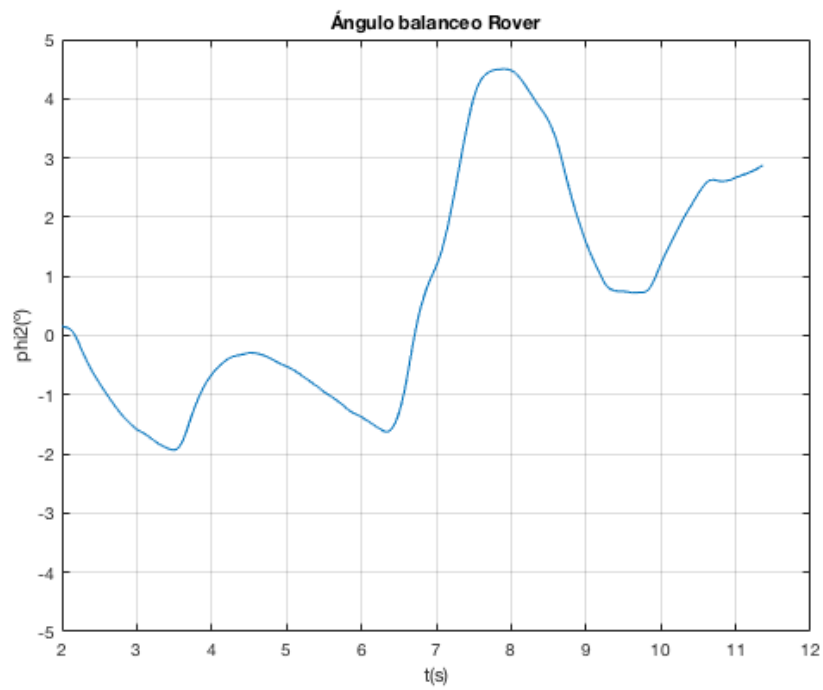
Además de este ángulo, se puede observar el comportamiento de los ángulos de cabeceo (Figura 17) y balanceo (figura 18). En este caso, cabe destacar el hecho de que inicialmente la trayectoria del vehículo se asemeja a una línea recta hasta que llega a un punto en el que comienza a girar, punto en el que el ángulo de cabeceo disminuye y el de balanceo aumenta, y gira hasta dar la vuelta completa. A su vez, el instante en el que el ángulo de balanceo es máximo, lo que indica que se encuentra perpendicular a la dirección de subida de la cuesta, coincide con el valor nulo del ángulo de cabeceo.

Esta trayectoria se puede comprobar con la representación de la trayectoria seguida por el Rover sobre el plano XY (figura 19).

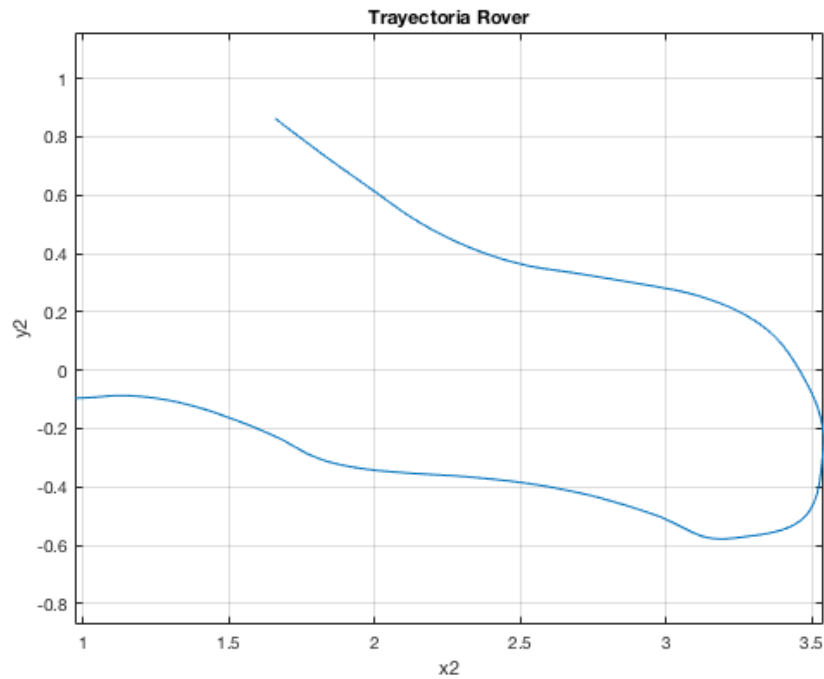




**Figura 19.** Ángulo cabeceo experimental

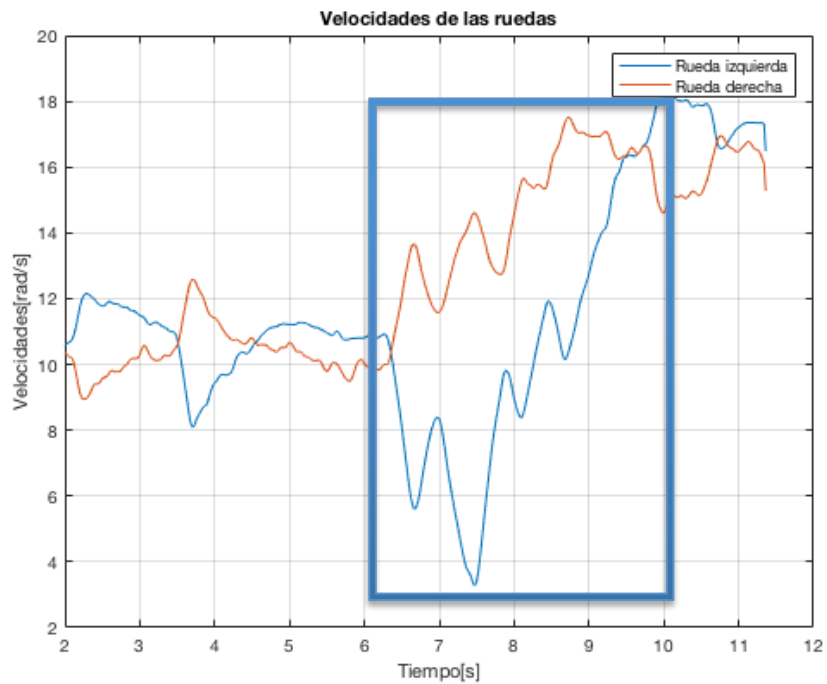


**Figura 20.** Ángulo balanceo experimental



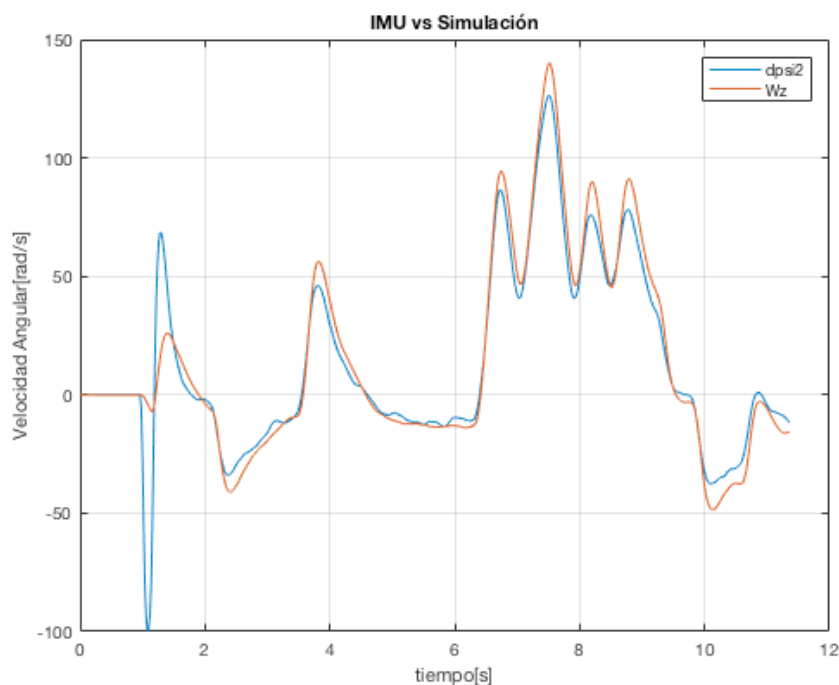
**Figura 21.** Trayectoria sobre el plano XY experimental

Este giro se debe a que entre en torno los segundos 6 a 10 la velocidad de la rueda derecha es ligeramente superior a la de la rueda izquierda como se puede observar en la figura 20.

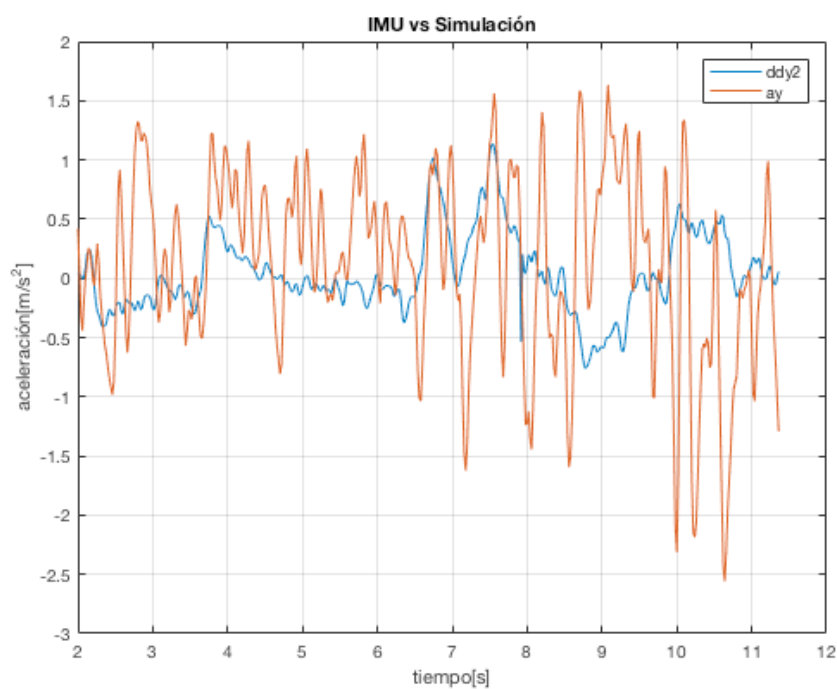


**Figura 22.** Velocidades angulares de las ruedas

Además de estas gráficas, se ha representado la comparativa entre la componente z de la velocidad angular registrada por el giroscopio y los valores calculados de la velocidad generalizada  $\dot{\psi}_2$ , valores que deberían asimilarse lo más posible. Así como la comparativa entre las aceleraciones en dirección y registradas por el IMU con las calculadas con el programa. En este caso, las diferencias se deben a tanto el ruido del sistema como las vibraciones introducidas por el terreno.



**Figura 23.** Velocidad angular IMU frente a la calculada



**Figura 24.** Aceleración IMU frente calculada

## 6 CONCLUSIONES Y DESARROLLOS FUTUROS

Las posibilidades de simulación del vehículo utilizado en este trabajo era reducidas dado que los grados de libertad de los que disponía el sistema eran los giros de las ruedas, que todos los sólidos estaban conectados rígidamente y las ruedas motoras no podían girar de forma independiente al chasis. Sin embargo, este sistema nos ha brindado la oportunidad de simular el movimiento del vehículo en tres dimensiones y de mejorar el modelo anterior que solo permitía el movimiento en dos dimensiones tal y como se ha explicado con anterioridad. De modo que siempre que sea posible parametrizar la superficie por la que queremos que el Rover circule, la simulación de su movimiento es posible.

Previo a variaciones del modelo de vehículo utilizado, se podrían realizar variaciones en las superficies utilizadas para la simulación del rover y de esta forma simular su comportamiento ante un mayor número de casos.

En cuanto a variaciones del modelo del rover, inicialmente se podría realizar la simulación de un vehículo virtual formado por cuatro ruedas, siendo las dos delanteras motoras también, pero que en este caso se podría controlar el giro de las ruedas delanteras independientemente, sin necesidad de introducir una mayor velocidad angular a una de ellas. En este caso, el sistema también dispondría de dos grados de libertad, pero repartidos entre velocidad de las ruedas y dirección del conjunto formado por ambas ruedas, asimilándose así más al comportamiento real de un vehículo en el que controlas la velocidad con los pedales y la dirección con el volante.

Si este vehículo Rover 2.0 fuese creado físicamente, se tendrían que hacer variaciones. Las más importantes son: En primer lugar cambiar la rueda trasera y el sólido que la une con el chasis, por dos ruedas de igual tamaño a las delanteras las cuales carecerían de la posibilidad de girar sobre el eje Z global. En segundo lugar, tampoco se necesitarían dos motores eléctricos, uno por rueda, dado que en este caso ambas girarían a la misma velocidad. Se tendría que añadir, un sólido que permitiera el movimiento relativo del conjunto de las dos ruedas delanteras respecto al chasis, y añadir otro pequeño motor eléctrico que permitiera que el giro fuese controlado, siendo entonces este giro uno de los grados de libertad del sistema. Mediante el giroscopio ya instalado sobre el Rover anterior podríamos registrar los giros realizados por el vehículo e introducir en el programa de simulación estos valores como una de las coordenadas independientes.

En cuanto al programa encargado de hacer la simulación, tanto el número de coordenadas como el número de ecuaciones de restricción cambiarían, manteniendo, como se ha dicho con anterioridad, el número de grados de libertad. De esta forma se modifica el número de componentes de cada una de las matrices que se han utilizado a lo largo de todo el sistema resolutivo, siendo mayor en este caso debido al aumento del número de ruedas existentes que modifica tanto el número de ecuaciones debido a la rodadura sin deslizamiento como las ecuaciones por el contacto rígido existente con la superficie de la carretera.

## 7 ANEXOS

### 7.1 Anexo A. Cálculo simbólico.

```
%calculo simbolico rover-carretera

clear
tic

syms x2 y2 z2 psi2 tet2 phi2 tet3 s3 tet4 s4 psi5 tet6 s6 scpx scpy scqx scqy scsx
scsy real
syms dx2 dy2 dz2 dpsi2 dtet2 dphi2 dtet3 ds3 dtet4 ds4 dpsi5 dtet6 ds6 dscpx dscpy
dscqx dscqy dscsx dscsy real

syms Rf Rr dx3 dy3 dz3 dx5 dx6 dz6 tetc real

syms m2 m3 m4 m5 m6 m7 g real
syms I2xx I2yy I2zz I2xy I2xz I2yz real
syms I3xx I3yy I3zz I3xy I3xz I3yz real
syms I4xx I4yy I4zz I4xy I4xz I4yz real
syms I5xx I5yy I5zz I5xy I5xz I5yz real
syms I6xx I6yy I6zz I6xy I6xz I6yz real
syms I7xx I7yy I7zz I7xy I7xz I7yz real

syms ParM3 ParM4 real

% Vector de coordenadas generalizadas
q = [ x2 y2 z2 psi2 tet2 phi2 tet3 s3 tet4 s4 psi5 tet6 s6 scpx scpy scqx scqy scsx
scsy]';
dq = [ dx2 dy2 dz2 dpsi2 dphi2 dtet2 dtet3 ds3 dtet4 ds4 dpsi5 dtet6 ds6 dscpx dscpy
dscqx dscqy dscsx dscsy]';

% Matrices de orientación
A2_1(1,:) = [cos(psi2) -sin(psi2) 0];
A2_1(2,:) = [sin(psi2) cos(psi2) 0];
A2_1(3,:) = [ 0 0 1];

A2_2(1,:) = [ cos(tet2) 0 -sin(tet2)];
A2_2(2,:) = [ 0 1 0 ];
A2_2(3,:) = [sin(tet2) 0 cos(tet2)];

A2_3(2,:) = [0 cos(phi2) sin(phi2)];
A2_3(3,:) = [0 -sin(phi2) cos(phi2)];
A2_3(1,:) = [1 0 0];

A24(1,:) = [ cos(tet4) 0 sin(tet4)];
A24(2,:) = [ 0 1 0];
A24(3,:) = [-sin(tet4) 0 cos(tet4)];

A23(1,:) = [ cos(tet3) 0 sin(tet3)];
A23(2,:) = [ 0 1 0];
A23(3,:) = [-sin(tet3) 0 cos(tet3)];

A25(1,:) = [cos(psi5) -sin(psi5) 0];
A25(2,:) = [sin(psi5) cos(psi5) 0];
A25(3,:) = [ 0 0 1];

A56(1,:) = [ cos(tet6) 0 sin(tet6)];
A56(2,:) = [ 0 1 0];
A56(3,:) = [-sin(tet6) 0 cos(tet6)];
```

```

A7(1,:) = [ cos(tetc) 0 -sin(tetc)];
A7(2,:) = [          0 1          0];
A7(3,:) = [sin(tetc) 0 cos(tetc)];

A2=A2_1*A2_2*A2_3;
A3 = A2*A23;
A4 = A2*A24;
A5 = A2*A25;
A6 = A5*A56;

%matlabFunction(A2,'file', 'RotMatSolido2');
%matlabFunction(A3,'file', 'RotMatSolido3');
%matlabFunction(A4,'file', 'RotMatSolido4');
%matlabFunction(A5,'file', 'RotMatSolido5');
%matlabFunction(A6,'file', 'RotMatSolido6');

dA2 =
diff(A2,psi2)*dpsi2+diff(A2,tet2)*dtet2+diff(A2,phi2)*dphi2+diff(A2,tet3)*dtet3+diff(A2,tet
4)*dtet4+diff(A2,psi5)*dpsi5+diff(A2,tet6)*dtet6;
dA3 =
diff(A3,psi2)*dpsi2+diff(A3,tet2)*dtet2+diff(A3,phi2)*dphi2+diff(A3,tet3)*dtet3+diff(A3,tet
4)*dtet4+diff(A3,psi5)*dpsi5+diff(A3,tet6)*dtet6;
dA4 =
diff(A4,psi2)*dpsi2+diff(A4,tet2)*dtet2+diff(A4,phi2)*dphi2+diff(A4,tet3)*dtet3+diff(A4,tet
4)*dtet4+diff(A4,psi5)*dpsi5+diff(A4,tet6)*dtet6;
dA5 =
diff(A5,psi2)*dpsi2+diff(A5,tet2)*dtet2+diff(A5,phi2)*dphi2+diff(A5,tet3)*dtet3+diff(A5,tet
4)*dtet4+diff(A5,psi5)*dpsi5+diff(A5,tet6)*dtet6;
dA6 =
diff(A6,psi2)*dpsi2+diff(A6,tet2)*dtet2+diff(A6,phi2)*dphi2+diff(A6,tet3)*dtet3+diff(A6,tet
4)*dtet4+diff(A6,psi5)*dpsi5+diff(A6,tet6)*dtet6;

% Posición en locales de puntos característicos

uG3 = [dx3 dy3 -dz3]';
uG4 = [dx3 -dy3 -dz3]';
uG5 = [dx3-dx5 0 (Rf+dz3)-(Rr+dz6)]';
uG6 = [-dx6 0 -dz6]';

uP = Rf*[sin(tet3) 0 -cos(tet3)]';
uQ = Rf*[sin(tet4) 0 -cos(tet4)]';
uS = Rr*[sin(tet6) 0 -cos(tet6)]';

%Posiciones de los solidos en globales
R2=[x2 y2 z2]';
R3 = R2 + A2*uG3;
R4 = R2 + A2*uG4;
R5 = R2 + A2*uG5;
R6 = R5 + A5*uG6;
R7=[0 0 0]';

%Velocidades angulares en locales
w4Lsk = simplify(A4'*dA4);
w4L = [-w4Lsk(2,3) w4Lsk(1,3) -w4Lsk(1,2)]';

w3Lsk = simplify(A3'*dA3);
w3L = [-w3Lsk(2,3) w3Lsk(1,3) -w3Lsk(1,2)]';

w2Lsk = simplify(A2'*dA2);
w2L = [-w2Lsk(2,3) w2Lsk(1,3) -w2Lsk(1,2)]';

w5Lsk = simplify(A5'*dA5);
w5L = [-w5Lsk(2,3) w5Lsk(1,3) -w5Lsk(1,2)]';

```

```

w6Lsk = simplify(A6'*dA6);
w6L = [-w6Lsk(2,3) w6Lsk(1,3) -w6Lsk(1,2)]';

% Traslaciones rotacionales
H2 = jacobian(R2,q); h2 = jacobian(H2*dq,q);
H3 = jacobian(R3,q); h3 = jacobian(H3*dq,q);
H4 = jacobian(R4,q); h4 = jacobian(H4*dq,q);
H5 = jacobian(R5,q); h5 = jacobian(H5*dq,q);
H6 = jacobian(R6,q); h6 = jacobian(H6*dq,q);

G2 = (jacobian(w2L,dq)); g2 = (jacobian(w2L,q));
G3 = (jacobian(w3L,dq)); g3 = (jacobian(w3L,q));
G4 = (jacobian(w4L,dq)); g4 = (jacobian(w4L,q));
G5 = (jacobian(w5L,dq)); g5 = (jacobian(w5L,q));
G6 = (jacobian(w6L,dq)); g6 = (jacobian(w6L,q));

%matlabFunction(G3,'file', 'MatrizG3' );
%matlabFunction(G4,'file', 'MatrizG4' );
%Restricciones contacto posicion

ucp=[scpx scpy 0]';
ucq=[scqx scqy 0]';
ucs=[scsx scsy 0]';

Rcp=R7+A7*ucp;
Rcq=R7+A7*ucq;
Rcs=R7+A7*ucs;

u3P = Rf*[sin(s3) 0 -cos(s3)]';
u4Q = Rf*[sin(s4) 0 -cos(s4)]';
u6S = Rr*[sin(s6) 0 -cos(s6)]';

R3p=R3+A3*u3P;
R4q=R4+A4*u4Q;
R6s=R6+A6*u6S;

%restricciones contacto normales

tcpx=jacobian(Rcp,scpx);
tcpy=jacobian(Rcp,scpy);
Ncp=simplify(cross(tcpx,tcpy));

tcqx=jacobian(Rcq,scqx);
tcqy=jacobian(Rcq,scqy);
Ncq=simplify(cross(tcqx,tcqy));

tcsx=jacobian(Rcs,scsx);
tcsy=jacobian(Rcs,scsy);
Ncs=simplify(cross(tcsx,tcsy));

t3p=jacobian(R3p,s3);
t2=[0 1 0]';
N3p=simplify(cross(t3p,t2));

t4q=jacobian(R4q,s4);
t2=[0 1 0]';
N4q=simplify(cross(t4q,t2));

t6s=jacobian(R6s,s6);
t2=[0 1 0]';
N6s=simplify(cross(t6s,t2));

Ch=[Rcp(1)-R3p(1) Rcp(2)-R3p(2) Rcp(3)-R3p(3) dot(Ncp,t3p) Rcq(1)-R4q(1) Rcq(2)-R4q(2)
Rcq(3)-R4q(3) dot(Ncq,t4q) Rcs(1)-R6s(1) Rcs(2)-R6s(2) Rcs(3)-R6s(3) dot(Ncs,t6s)]';
Cqh=jacobian(Ch,q);

```

```

dCqh = jacobian(Cqh*dq,q);
%matlabFunction(Ch,'file','MatrizCh');
%matlabFunction(Cqh,'file','MatrizCqh');
%matlabFunction(dCqh,'file','MatrizdCqh');

% Velocidades de los centros de gravedad
V2 = H2*dq;
V3 = H3*dq;
V4 = H4*dq;
V5 = H5*dq;
V6 = H6*dq;

% Restricciones de rodadura sin deslizamiento
VP = simplify(V3 + A3*cross(w3L,uP));
VQ = simplify(V4 + A4*cross(w4L,uQ));
VS = simplify(V6 + A6*cross(w6L,uS));

VPx=dot(VP,tcpx/norm(tcpx));
VPy=dot(VP,tcpy/norm(tcpy));
VQx=dot(VQ,tcqx/norm(tcqx));
VSx=dot(VS,tcsx/norm(tcsx));
VSy=dot(VS,tcsy/norm(tcsy));

Cnh = [VPx VPy VQx VSx VSy]';

B = jacobian(Cnh,dq);
dB = jacobian(B*dq,q);
%matlabFunction(Cnh,'file','MatrizCnh');
%matlabFunction(B,'file','MatrizB');
%matlabFunction(dB,'file','MatrizdB');

%Matriz restricciones
D=[Cqh;B];
dD=[dCqh;dB];

%matlabFunction(D,'file','MatrizD');
%matlabFunction(dD,'file','MatrizdD');

% Matrices de masas e inercias
M2 = m2*eye(3);
M3 = m3*eye(3);
M4 = m4*eye(3);
M5 = m5*eye(3);
M6 = m6*eye(3);

I2(1,:) = [I2xx I2xy I2xz];
I2(2,:) = [I2xy I2yy I2yz];
I2(3,:) = [I2xz I2yz I2zz];

I3(1,:) = [I3xx I3xy I3xz];
I3(2,:) = [I3xy I3yy I3yz];
I3(3,:) = [I3xz I3yz I3zz];

I4(1,:) = [I4xx I4xy I4xz];
I4(2,:) = [I4xy I4yy I4yz];
I4(3,:) = [I4xz I4yz I4zz];

I5(1,:) = [I5xx I5xy I5xz];
I5(2,:) = [I5xy I5yy I5yz];
I5(3,:) = [I5xz I5yz I5zz];

I6(1,:) = [I6xx I6xy I6xz];
I6(2,:) = [I6xy I6yy I6yz];
I6(3,:) = [I6xz I6yz I6zz];

```



```

Z0 = zeros(3,3);
MM(1:3,:) = [M2 Z0 Z0 Z0 Z0 Z0 Z0 Z0 Z0 Z0];
MM(4:6,:) = [Z0 M3 Z0 Z0 Z0 Z0 Z0 Z0 Z0 Z0];
MM(7:9,:) = [Z0 Z0 M4 Z0 Z0 Z0 Z0 Z0 Z0 Z0];
MM(10:12,:) = [Z0 Z0 Z0 M5 Z0 Z0 Z0 Z0 Z0 Z0];
MM(13:15,:) = [Z0 Z0 Z0 Z0 M6 Z0 Z0 Z0 Z0 Z0];
MM(16:18,:) = [Z0 Z0 Z0 Z0 Z0 I2 Z0 Z0 Z0 Z0];
MM(19:21,:) = [Z0 Z0 Z0 Z0 Z0 Z0 I3 Z0 Z0 Z0];
MM(22:24,:) = [Z0 Z0 Z0 Z0 Z0 Z0 Z0 I4 Z0 Z0];
MM(25:27,:) = [Z0 Z0 Z0 Z0 Z0 Z0 Z0 Z0 I5 Z0];
MM(28:30,:) = [Z0 Z0 Z0 Z0 Z0 Z0 Z0 Z0 Z0 I6];

% Fuerzas generalizadas cuadráticas en velocidad
Z0 = zeros(3,1);
QQv(16:18,:) = -cross(w2L,I2*w2L);
QQv(19:21,:) = -cross(w3L,I3*w3L);
QQv(22:24,:) = -cross(w4L,I4*w4L);
QQv(25:27,:) = -cross(w5L,I5*w5L);
QQv(28:30,:) = -cross(w6L,I6*w6L);
QQv(1:3,:) = Z0;
QQv(4:6,:) = Z0;
QQv(7:9,:) = Z0;
QQv(10:12,:) = Z0;
QQv(13:15,:) = Z0;

% Fuerzas gravitatorias
QQgrav(1:3,1) = [0 0 -m2*g]';
QQgrav(4:6,1) = [0 0 -m3*g]';
QQgrav(7:9,1) = [0 0 -m4*g]';
QQgrav(10:12,1) = [0 0 -m5*g]';
QQgrav(13:15,1) = [0 0 -m6*g]';
QQgrav(16:30,1) = zeros(15,1);

% Matrices de transformación cinemática
L = [H2' H3' H4' H5' H6' G2' G3' G4' G5' G6']';
l = [h2' h3' h4' h5' h6' g2' g3' g4' g5' g6']';

M = L'*MM*L;
Qv = L'*(QQv-MM*l*dq);
Qgrav = L'*QQgrav;

Qmot = G3'*[0 ParM3 0]' + G4'*[0 ParM4 0]';

MatMasaFuerzas = [M Qv Qgrav Qmot];

%matlabFunction(MatMasaFuerzas,'file', 'MatMasaFuerzas');

toc

```

## 7.2 Anexo B. Simulación Dinámica Inversa

```
%SIMULACION DINAMICA
clear;clc
tic
global Param
Param = ParametrosRover;

Ind = Param.Ind;
Dep = Param.Dep;
nn = Param.nn;
tetc= Param.tetc;

%EXTRACCION DATOS
load DATOS_1.mat;
t      = DATOS_1(:,1);
tet3   = DATOS_1(:,2);
dtet3  = DATOS_1(:,3);
ddtet3 = DATOS_1(:,4);
tet4   = DATOS_1(:,5);
dtet4  = DATOS_1(:,6);
ddtet4 = DATOS_1(:,7);

wx      = DATOS_1(:,8);
wy      = DATOS_1(:,9);
wz      = DATOS_1(:,10);
ax      = DATOS_1(:,11);
ay      = DATOS_1(:,12);
az      = DATOS_1(:,13);

%CONDICIONES INICIALES 1
At = (t(end) - t(end-1));

ci=condicionesIniciales(tetc);

q0 = [0 tetc 0 0 0 0 0.2 0.08 0.2 -0.08 0 0]';

[q_0,fval]=fsolve(@MatrizCh,q0,optimoptions('fsolve','Display','off'));

x2_0    = ci(1);
y2_0    = q_0(1);
z2_0    = ci(2);
psi2_0  = 0;
tet2_0  = tetc;
phi2_0  = 0;
s3_0    = q_0(3);
s4_0    = q_0(4);
psi5_0  = 0;
tet6_0  = 0;
s6_0    = q_0(5);
scpx_0  = q_0(6);
scpy_0  = q_0(7);
scqx_0  = q_0(8);
scqy_0  = q_0(9);
scsx_0  = q_0(10);
scsy_0  = q_0(11);

qdep_0  = [x2_0 y2_0 z2_0 psi2_0 tet2_0 phi2_0 s3_0 s4_0 psi5_0 tet6_0 s6_0 scpx_0
scpy_0 scqx_0 scqy_0 scsx_0 scsy_0]';
q       = zeros(nn,length(t));
dq      = zeros(nn,length(t));
```

```

ddq = zeros(nn,length(t));

ParM3 = zeros(length(t),1);
ParM4 = zeros(length(t),1);

for i = 1 : length(t)

    %DATOS COORDENADAS INDEPENDIENTES
    qind = [tet3(i) tet4(i)]';
    dqind = [dtet3(i) dtet4(i)]';
    ddqind = [ddtet3(i) ddtet4(i)]';

    if t(i) == 0.0
        qdep = qdep_0;
    end

    q(:,i) = ReordenaCoord(qind,qdep);

    %CALCULO MATRIZ D
    D = MatrizD(q(:,i),tetc);
    Dind = D(:,Ind);
    Ddep = D(:,Dep);

    %CALCULO VELOCIDADES DEPENDIENTES
    dqdep=-Ddep\Dind*dqind;

    dq(:,i) = ReordenaCoord(dqind,dqdep);

    %Ch=MatrizCh(q(:,i),tetc);
    %Cnh=MatrizCnh(q(:,i),dq(:,i),tetc);

    %CALCULO MATRIZ dD
    dD = MatrizdD(q(:,i),dq(:,i),tetc);

    %CALCULO MATRIZ DE MASA Y FUERZAS GENERALIZADAS
    MMF = MatMasaFuerzas(q(:,i),dq(:,i),0,0);
    M = MMF(:,1:nn);
    Qv = MMF(:,nn+1);
    Qgrav = MMF(:,nn+2);

    %CALCULO MATRIZ FUERZAS MOTORAS
    G3 = MatrizG3(q(:,i));
    G4 = MatrizG4(q(:,i));

    S = [G3(2,:) ' G4(2,:) '];

    M_dep = [M(:,1) M(:,2) M(:,3) M(:,4) M(:,5) M(:,6) M(:,8) M(:,10) M(:,11) M(:,12)
M(:,13) M(:,14) M(:,15) M(:,16) M(:,17) M(:,18) M(:,19)];
    M_ind = [M(:,7) M(:,9)];

    D_dep = [D(:,1) D(:,2) D(:,3) D(:,4) D(:,5) D(:,6) D(:,8) D(:,10) D(:,11) D(:,12)
D(:,13) D(:,14) D(:,15) D(:,16) D(:,17) D(:,18) D(:,19)];
    D_ind = [D(:,7) D(:,9)];

    %RESOLUCION Ax=b
    AA = zeros(36,36);
    bb = zeros(36,1);

    AA(1:19,1:17) = M_dep;
    AA(1:19,18:19) = -S;
    AA(1:19,20:36) = D';

    AA(20:36,1:17) = D_dep;
    AA(20:36,18:19) = zeros(17,2);
    AA(20:36,20:36) = zeros(17,17);

    bb(1:19) = -M_ind*ddqind + Qgrav + Qv;

```

```

bb(20:36) = -dD*dq(:,i) - D_ind*ddqind;

x=AA\bb;

ParM3(i) = x(18);
ParM4(i) = x(19);

Lambdas(:,i) = x(20:36);

ddqdep = x(1:17,1);

%VELOCIDADES Y ACELERACIONES
dx2 = dqdep(1); ddx2 = ddqdep(1);
dy2 = dqdep(2); ddy2 = ddqdep(2);
dz2 = dqdep(3); ddz2 = ddqdep(3);
dpsi2 = dqdep(4); ddpsi2 = ddqdep(4);
dtet2 = dqdep(5); ddtet2 = ddqdep(5);
dphi2 = dqdep(6); ddphi2 = ddqdep(6);
ds3 = dqdep(7); dds3 = ddqdep(7);
ds4 = dqdep(8); dds4 = ddqdep(8);
dpsi5 = dqdep(9); ddpsi5 = ddqdep(9);
dtet6 = dqdep(10); ddtet6 = ddqdep(10);
ds6 = dqdep(11); dds6 = ddqdep(11);
dscpx = dqdep(12); ddscpx = ddqdep(12);
dscpy = dqdep(13); ddscpy = ddqdep(13);
dscqx = dqdep(14); ddscqx = ddqdep(14);
dscqy = dqdep(15); ddscqy = ddqdep(15);
dscsx = dqdep(16); ddscsx = ddqdep(16);
dscsy = dqdep(17); ddscsy = ddqdep(17);

%INTEGRACION NUMERICA
x2 = x2_0 + dx2*At + 0.5*At^2*ddx2;
y2 = y2_0 + dy2*At + 0.5*At^2*ddy2;
z2 = z2_0 + dz2*At + 0.5*At^2*ddz2;
psi2 = psi2_0 + dpsi2*At + 0.5*At^2*ddpsi2;
tet2 = tet2_0 + dtet2*At + 0.5*At^2*ddtet2;
phi2 = phi2_0 + dphi2*At + 0.5*At^2*ddphi2;
s3 = s3_0 + ds3*At + 0.5*At^2*dds3;
s4 = s4_0 + ds4*At + 0.5*At^2*dds4;
psi5 = psi5_0 + dpsi5*At + 0.5*At^2*ddpsi5;
tet6 = tet6_0 + dtet6*At + 0.5*At^2*ddtet6;
s6 = s6_0 + ds6*At + 0.5*At^2*dds6;
scpx = scpx_0 + dscpx*At + 0.5*At^2*ddscpx;
scpy = scpy_0 + dscpy*At + 0.5*At^2*ddscpy;
scqx = scqx_0 + dscqx*At + 0.5*At^2*ddscqx;
scqy = scqy_0 + dscqy*At + 0.5*At^2*ddscqy;
scsx = scsx_0 + dscsx*At + 0.5*At^2*ddscsx;
scsy = scsy_0 + dscsy*At + 0.5*At^2*ddscsy;

%CAMBIO VALORES INICIALES
x2_0 = x2;
y2_0 = y2;
z2_0 = z2;
psi2_0 = psi2;
tet2_0 = tet2;
phi2_0 = phi2;
s3_0 = s3;
s4_0 = s4;
psi5_0 = psi5;
tet6_0 = tet6;
s6_0 = s6;
scpx_0 = scpx;
scpy_0 = scpy;
scqx_0 = scqx;
scqy_0 = scqy;
scsx_0 = scsx;
scsy_0 = scsy;

q(:,i) = [x2 y2 z2 psi2 tet2 phi2 tet3(i) s3 tet4(i) s4 psi5 tet6 s6 scpx scpy

```

```
scqx scqy scsx scsy]';  
  
    qdep = [x2 y2 z2 psi2 tet2 phi2 s3 s4 psi5 tet6 s6 scpx scpy scqx scqy scsx  
scsy]';  
end  
  
%Animacion  
AnimaRover(q,tetc)
```

### 7.3 Anexo C. AnimaRover

```

function AnimaRover(q,tetc)

global Param

Rf = Param.Rf;
Rr = Param.Rr;

Lar = Param.Lar;
Anc = Param.Anc;
Alt = Param.Alt;

cdgx = Param.cdgx;
cdgz = Param.cdgz;

C1 = Param.C1;
C2 = Param.C2;
C3 = Param.C3;
C4 = Param.C4;

% Chasis Rover
uAA1bar = [Lar-cdgx  Anc/2  cdgz]';
uAA2bar = [Lar-cdgx -Anc/2  cdgz]';
uAA3bar = [-cdgx      -Anc/2  cdgz]';
uAA4bar = [-cdgx      Anc/2  cdgz]';

uBB1bar = [-cdgx      C3/2  cdgz]';
uBB2bar = [-cdgx      -C3/2  cdgz]';
uBB3bar = [-cdgx-C3   -C3/2  cdgz]';
uBB4bar = [-cdgx-C3   C3/2  cdgz]';

uDD1bar = [Lar-cdgx  Anc/2  -cdgz]';
uDD2bar = [Lar-cdgx -Anc/2  -cdgz]';
uDD3bar = [-cdgx      -Anc/2  -cdgz]';
uDD4bar = [-cdgx      Anc/2  -cdgz]';

uEE1bar = [Lar-C1 -Anc/2      -cdgz]';
uEE2bar = [Lar-C1 -Anc/2-C2 -cdgz]';

uFF1bar = [Lar-C1 Anc/2      -cdgz]';
uFF2bar = [Lar-C1 Anc/2+C2 -cdgz]';

uGG2bar = [0 0 -(Alt+Rf-Rr)]';

alfa = 0 : pi/8 : 2*pi;

[~,col] = size(q);

for i = 1 : col

    A2 = RotMatSolido2(q(:,i));
    R2 = [q(1,i) q(2,i) q(3,i)]';

    % Rueda derecha
    u3bar = [Lar-C1 -Anc/2-C2 -cdgz]';
    R3 = R2 + A2*u3bar;
    A3 = RotMatSolido3(q(:,i));

    RP = zeros(3,18);

    for j = 1 : length(alfa)
        RP(:,j) = R3 + Rf*A3*[cos(alfa(j)) 0 sin(alfa(j))]'';
    end
end

```

```

end

RP(:,18) = R3;

% Rueda izquierda
u4bar = [Lar-C1 Anc/2+C2 -cdgz]';
R4 = R2 + A2*u4bar;
A4 = RotMatSólido4(q(:,i));

RQ = zeros(3,18);

for j = 1 : length(alfa)
    RQ(:,j) = R4 + Rf*A4*[cos(alfa(j)) 0 sin(alfa(j))]'';
end

RQ(:,18) = R4;

% Rueda trasera
A5 = RotMatSólido5(q(:,i));
u5bar = [-cdgx-C3/2 0 cdgz]';
R5 = R2 + A2*u5bar;

u6bar = [-C4 0 -(Alt+Rf-Rr)]';
R6 = R5 + A5*u6bar;
A6 = RotMatSólido6(q(:,i));

RS = zeros(3,18);

for j = 1 : length(alfa)
    RS(:,j) = R6 + Rr*A6*[cos(alfa(j)) 0 sin(alfa(j))]'';
end

RS(:,18) = R6;

%Ploteo Chasis Rover
RAA1 = R2 + A2*uAA1bar;
RAA2 = R2 + A2*uAA2bar;
RAA3 = R2 + A2*uAA3bar;
RAA4 = R2 + A2*uAA4bar;

RBB1 = R2 + A2*uBB1bar;
RBB2 = R2 + A2*uBB2bar;
RBB3 = R2 + A2*uBB3bar;
RBB4 = R2 + A2*uBB4bar;

RDD1 = R2 + A2*uDD1bar;
RDD2 = R2 + A2*uDD2bar;
RDD3 = R2 + A2*uDD3bar;
RDD4 = R2 + A2*uDD4bar;

REE1 = R2 + A2*uEE1bar;
REE2 = R2 + A2*uEE2bar;
RFF1 = R2 + A2*uFF1bar;
RFF2 = R2 + A2*uFF2bar;

RGG2 = R5 + A5*uGG2bar;

Chasis = [RAA1 RAA2 RAA3 RAA4 RAA1 RDD1 RDD2 RAA2 RDD2 RDD3 RAA3...
          RDD3 RDD4 RAA4 RDD4 RDD1];

x = Chasis(1,:);
y = Chasis(2,:);
z = Chasis(3,:);

figure(1000)
grid on
plot3(x,y,z, 'y');
hold on

```

```

clear x y z

SubChasis = [RBB1 RBB2 RBB3 RBB4 RBB1];

x = SubChasis(1,:);
y = SubChasis(2,:);
z = SubChasis(3,:);

plot3(x,y,z,'y');
clear x y z

EjeDerecho = [REE1 REE2];

x = EjeDerecho(1,:);
y = EjeDerecho(2,:);
z = EjeDerecho(3,:);

plot3(x,y,z,'y');
clear x y z

EjeIzquierdo = [RFF1 RFF2];

x = EjeIzquierdo(1,:);
y = EjeIzquierdo(2,:);
z = EjeIzquierdo(3,:);

plot3(x,y,z,'y');
clear x y z

% Rueda Derecha
x = RP(1,:);
y = RP(2,:);
z = RP(3,:);

plot3(x,y,z,'r')
clear x y z

% Rueda Izquierda
x = RQ(1,:);
y = RQ(2,:);
z = RQ(3,:);

plot3(x,y,z,'r')
clear x y z

% Eje Rueda Jockey
Eje = [R5 RGG2 R6 R5];

x = Eje(1,:);
y = Eje(2,:);
z = Eje(3,:);

plot3(x,y,z,'m')
clear x y z

% Rueda Trasera
x = RS(1,:);
y = RS(2,:);
z = RS(3,:);

plot3(x,y,z,'r')
clear x y z

%carretera
teta=tetc;
s=0:0.1:(max(q(1,:))/cos(teta))+1;

```



```
y=-max(abs(q(2,:)))-1:0.1:max(abs(q(2,:)))+1;
[S,Y]=meshgrid(s,y);
x=S*cos(teta);
z=S*sin(teta);
surf(x,Y,z)
clear x y z

grid on
hold off

siz = 0.8;
axis([R2(1)-siz R2(1)+siz R2(2)-siz R2(2)+siz cdgz-siz cdgz+siz])
pause(0.03)

end
```

## 7.4 Anexo D. Registro de Datos

```

%Cargamos los datos
load DAT_V89.txt;
t=DAT_V89(:,1);
ax=DAT_V89(:,2);
ay=DAT_V89(:,3);
az=DAT_V89(:,4);
wx=DAT_V89(:,5);
wy=DAT_V89(:,6);
wz=DAT_V89(:,7);
pulsosR=-DAT_V89(:,8);
pulsosL= DAT_V89(:,9);

%desechamos valores hasta t=1
t=t-1;
t=t(51:end);
ax=ax(51:end);
ay=ay(51:end);
az=az(51:end);
wx=wx(51:end);
wy=wy(51:end);
wz=wz(51:end);
pulsosR=pulsosR(51:end);
pulsosL=pulsosL(51:end);

%Convertimos pulsos a radianes
tet4=(pulsosR/960)*2*pi;
tet3=(pulsosL/960)*2*pi;

At=t(2)-t(1);    %periodo de adquisicion
fadq=1/At;       %frecuencia de adquisicion
fnyq=0.5*fadq;   %frecuencia de nyquist

%filtrado de posiciones
[b,a] = butter(2,4/fadq,'low');
tet3f=filter(b,a,tet3);
tet4f=filter(b,a,tet4);

%Calculo de velocidades y aceleraciones
dtet3=DerivadaPrimera(t,tet3);
dtet4=DerivadaPrimera(t,tet4);
ddtet3=DerivadaSegunda(t,tet3);
ddtet4=DerivadaSegunda(t,tet4);

%filtrado velocidades y aceleraciones
dtet3f=filter(b,a,dtet3);
dtet4f=filter(b,a,dtet4);
ddtet3f=filter(b,a,ddtet3);
ddtet4f=filter(b,a,ddtet4);

%filtrado de a y w
axf=filter(b,a,ax);
ayf=filter(b,a,ay);
azf=filter(b,a,az);
wxf=filter(b,a,wx);
wyf=filter(b,a,wy);
wzf=filter(b,a,wz);

DATOS_1=[t tet3f dtet3f' ddtet3f' tet4f dtet4f' ddtet4f' wxf wyf wzf axf ayf azf];
save DATOS_1.mat

```

## 8 BIBLIOGRAFÍA

---

- [1] D. Schramm, M. Hiller, and R. Bardini, *Vehicle dynamics: Modeling and simulation*. Springer, 2014.
- [2] J. Wittenburg, *Dynamics of Multibody Systems second edition*. Springer, 2008.
- [3] J. L. Escalona, “Dinámica de sistemas multicuerpo.” Apuntes de la asignatura de Cinemática y Dinámica de Máquinas, 2017.
- [4] OpenCourseWare, “Análisis Dinámico de Sistemas Multicuerpo,” Universidad Politécnica de Madrid, 2005.
- [5] I. Mula Vivero, “Estudio del comportamiento dinámico de un vehículo utilizando la herramienta simmechanics de matlab,” Universidad Carlos III, 2009.
- [6] J. Domínguez Abascal, Ed., *Teoría de Máquinas y Mecanismos*. Editorial Universidad de Sevilla, 2016.
- [7] P. Acuña León, “Adecuación y simulación cinemática y dinámica de un vehículo a control remoto para las prácticas de la asignatura de Cinemática y Dinámica de Máquinas,” Universidad de Sevilla, 2017.
- [8] A. A. Shabana, *Computational Dynamics, Third Edition*. Wiley, 2009.
- [9] J. Domínguez Abascal, E. Reina Romo, and R. Chamorro Moreno, “Análisis en frecuencia de señales temporales.” Apuntes de la asignatura de Ampliación de Cinemática y Dinámica de Máquinas. Universidad de Sevilla.